

適応射影劣勾配法に基づく
時変グラフ学習アルゴリズムに関する研究

伊藤 楓馬

学籍番号: 62001901

指導教員: 湯川 正裕

慶應義塾大学 理工学部 電気情報工学科 4 年

2025 年 2 月 28 日

概要

時変グラフ構造推定とは、観測される信号からその信号が生成されるグラフ構造を推定する問題である。オンライン推定のフレームワークとして、予測-修正アルゴリズムを用いた手法が提案されている。しかし、グラフ構造が急変するような状況下においては、収束が遅くなるという問題がある。本論文では並列射影の最適化手法を用いて、予測-修正アルゴリズムを改善する手法を提案する。また数値実験により、提案法が従来法よりも精度が向上したことを示している。

謝辞

本研究を進めるにあたり、指導教員の湯川正裕教授には多くの指導を賜りました。ここに感謝の意を示します。また、本研究を完遂できたのは周囲の全ての方々のおかげです。研究室の先輩方、同輩の皆様、支えてくださった家族、友人、全ての皆様に深く感謝をいたします。

目次

概要	1
謝辞	2
第 1 章 導入	1
第 2 章 準備	2
2.1 表記法	2
2.2 グラフ構造推定	2
2.2.1 観測される信号	2
2.2.2 構造方程式モデル (Structural Equation Model: SEM)	2
2.3 予測-修正アルゴリズム	3
2.3.1 定式化	4
2.4 適応並列射影劣勾配法 (Adaptive Parallel Subgradient Projection: APSP)	6
第 3 章 提案法	7
3.1 アイデア	7
3.2 提案手法の優位性	7
3.3 提案手法の定式化	8
第 4 章 実験設定	9
4.1 シミュレーションパラメータ	9
4.2 初期隣接行列の生成	9
4.3 データ生成方法	10
4.4 アルゴリズムの初期化と実行	10
4.5 各アルゴリズムのパラメータ	10
4.6 評価指標	12
4.7 実験結果	13
4.7.1 グラフ構造が時不変な場合	13
4.7.2 時間方向のグラフ構造変化に対するアルゴリズムの性能評価	14
第 5 章 結論	16
参考文献	16

第 1 章 導入

近年、グラフ信号処理 (Graph Signal Processing: GSP) は、ソーシャルネットワーク、金融市場、脳ネットワーク、電力網、交通ネットワークなど、様々な分野で注目を集めている。GSP は、グラフ構造上に定義された信号 (グラフ信号) を解析する枠組みを提供し、その応用範囲はデータ解析や機械学習、最適化分野へと拡大している。グラフ構造はしばしば未知であるため、データからグラフ構造を「学習」すること、すなわちグラフトポロジー同定 (Graph Topology Identification: GTI) が重要な課題として認識されてきた。

従来の GTI は、固定 (時不変) のグラフ構造を想定していたが、多くの実世界ネットワークは時变的なふるまいを示す。たとえば、金融ネットワークでは市場の急激な変動や経済政策によってノード間の関係が日々変化し、脳ネットワークでは神経活動が時間とともに変動する。こうした非定常な環境下では、単純なバッチ処理による静的なグラフ推定手法では十分に対応できず、逐次的に変化を追跡できるオンラインな手法が求められる。

近年、時変グラフ構造同定 (Time-Varying Graph Topology Identification) に関する研究が進められており、非定常データに対してグラフ構造を漸次的に推定するための様々な手法が提案されている。これらの手法は、時系列データから得られる統計量 (共分散行列など) を更新しつつ、グラフ構造を漸次的に推定する枠組みを提供する。しかし、従来手法の多くは、グラフ構造が急激に変化する状況や、非線形な動態を有するシナリオでは収束が遅くなるといった問題が残されている。

特に、ガウス型のモデル仮定に基づく Gaussian Graphical Model (GGM) や、ノード間の線形関係を仮定する構造方程式モデル (Structural Equation Model: SEM)、そして平滑性に基づく Smoothness-Based Model (SBM) など、グラフ学習に用いられる各種モデルを時変環境で適用する際には、計算コストや収束速度が課題となる。さらに、オンライン処理が求められる実世界のアプリケーション (通信ネットワークでの異常検知、株式市場でのポートフォリオ最適化、脳波データ解析など) では、低遅延かつ適応的な推定アルゴリズムの実装が不可欠となる。

本研究では、このような時変グラフ構造のオンライン推定問題に対し、既存の予測-修正 (Prediction-Correction) フレームワークを拡張し、適応並列射影劣勾配法 (Adaptive Parallel Subgradient Projection: APSP) を統合した新たな手法を提案する。APSP は複数の制約集合に対する並列射影操作を通じて高速な収束性を達成する劣勾配ベースの手法であり、時変最適化問題への応用によって、急峻な変化に対しても適応的に動作可能である。本手法では、直近の少数データに基づく並列射影による修正ステップを導入し、過去に蓄積された大量のデータに依存せずとも、高精度かつ収束速度の優れたグラフ構造推定を達成することを目指す。

本論文の構成は以下のとおりである。第 2 章では本研究で用いる数学的準備と記法を示し、従来手法の問題点を整理する。第 3 章では提案手法のアイデアと定式化を詳述し、第 4 章で実験設定と数値実験を行い、提案法の有効性を示す。第 5 章で結論と今後の展望を述べる。

第2章 準備

2.1 表記法

以下、本論文で使用する表記法について説明する。

\mathbb{N} は非負整数の集合を表す。 \mathbb{N}_+ は \mathbb{N} からゼロを除いた集合、すなわち $\mathbb{N}_+ = \mathbb{N} \setminus \{0\}$ である。ベクトル \mathbf{x} の第 i 成分は $x(i)$ と記す。行列 \mathbf{X} の (i, j) 成分は $X(i, j)$ と表す。行列の転置は $^\top$ を用いて示し、行列の擬似逆行列は † で表す。行列のトレースは $\text{tr}(\cdot)$ で示し、行列をベクトル化する操作は $\text{vec}(\cdot)$ で表す。 $\mathbf{0}_N$ は全ての要素がゼロである N 次元ベクトルを、 $\mathbf{1}_N$ は全ての要素が1である N 次元ベクトルを表す。 \mathbf{I}_N は N 次元単位行列を示す。 \otimes はクロネッカー積を、 \odot はアダマール積（要素ごとの積）を、 \oslash はアダマール除算（要素ごとの除算）を表す。 \circ はアダマールべき乗（要素ごとのべき乗）を示す。 $[\cdot]_+$ は最大値を要素ごとに計算する操作であり、具体的には $[\mathbf{x}]_+ = [\max(0, x(1)), \max(0, x(2)), \dots, \max(0, x(n))]^\top$ を意味する。 $\iota_{\mathcal{X}}(\cdot)$ は凸集合 \mathcal{X} の指示関数であり、 $\mathbf{x} \in \mathcal{X}$ のとき $\iota_{\mathcal{X}}(\mathbf{x}) = 0$ 、それ以外の場合は $+\infty$ となる。 $f \circ g(\cdot)$ は関数 $f(\cdot)$ と $g(\cdot)$ の合成を示す。 $f(\mathbf{x}; t)$ は時間 t でパラメータ化された関数であり、引数は $\mathbf{x} \in \mathbb{R}^N$ である。 $\nabla_{\mathbf{x}} f(\mathbf{x}; t)$ は点 $(\mathbf{x}; t)$ における関数 $f(\mathbf{x}; t)$ の \mathbf{x} に関する勾配を表す。 $\nabla_{\mathbf{xx}} f(\mathbf{x}; t)$ は同じ点でのヘシアン（2次微分行列）を示す。 $\nabla_{t\mathbf{x}} f(\mathbf{x}; t)$ は勾配の時間微分であり、これは $\nabla_{\mathbf{x}} f(\mathbf{x}; t)$ の時間 t に関する偏微分である。これは目的関数の混合1次偏導関数ベクトルを示す。 $\|\cdot\|_p$ はベクトルの ℓ_p ノルム、または行列をベクトル化した場合の ℓ_p ノルムを表す。 $\|\cdot\|_F$ は行列のフロベニウスノルムを示す。添字のないノルム $\|\cdot\|$ はスペクトルノルムを表す。

2.2 グラフ構造推定

グラフ構造推定（Graph Topology Identification: GTI）とは、以下の (1) 式のような逆問題として定式化される [1]。

$$\mathbf{S} = \mathcal{F}^{-1}(\mathbf{X}) \quad (1) \quad \{\text{eq:gti}\}$$

ここで、 \mathbf{X} は観測された信号の行列、 \mathbf{S} はグラフ構造を表現する行列（隣接行列など）、 \mathbf{S} からデータ \mathbf{X} が生成される過程を $\mathcal{F}(\cdot)$ で表す。データの生成過程は $\mathbf{X} = \mathcal{F}(\mathbf{S})$ と表現される。

2.2.1 観測される信号

本資料内でのグラフ構造推定における観測されるデータについて説明する。時刻 t で観測する信号は N 個のノードそれぞれに対応する実数値であり、まとめてベクトル $\mathbf{x}_t \in \mathbb{R}^N$ で表現する。また T 回に渡って得られたデータをまとめて行列 $\mathbf{X} = [\mathbf{x}_1 \dots \mathbf{x}_T] \in \mathbb{R}^{N \times T}$ で表現する。

2.2.2 構造方程式モデル (Structural Equation Model: SEM)

構造方程式モデル（Structural Equation Model: SEM）とは観測される信号 \mathbf{x}_t とグラフ構造を表す隣接行列 \mathbf{S} の関係に対して、(2) 式のような線形モデルが成り立つと仮定をおくモデルである [1]。

$$x_t(i) = \sum_{j \neq i} S(i, j)x_t(j) + \varepsilon_t(i), \quad t = 1, \dots, T \quad (2) \quad \{\text{eq:sem_entry}\}$$

ここでノード数を N 、時刻 t での信号をベクトル $\mathbf{x}_t \in \mathbb{R}^N$ 、隣接行列を \mathbf{S} 、ガウスノイズを $\varepsilon_t \sim \mathcal{N}(\mathbf{0}_N, \sigma^2 \mathbf{I}_N)$ とする。また想定するグラフは無向グラフであり、自己ループが存在しないとする。このとき、隣接行列 \mathbf{S} は以下の条件を満たす。

$$\mathbf{S} \in \mathcal{S} := \{\mathbf{A} \in \mathbb{R}^{N \times N} \mid \text{diag}(\mathbf{A}) = \mathbf{0}, \mathbf{A} = \mathbf{A}^\top\} \quad (3) \quad \{\mathbf{s}\}$$

ここで (2) 式は、(4) 式のように表現できる。

$$\mathbf{x}_t = \mathbf{S} \mathbf{x}_t + \varepsilon_t, \quad t = 1, \dots, T \quad (4) \quad \{\text{eq:sem_matrix}\}$$

グラフ構造を表現する隣接行列 \mathbf{S} が時不変である場合の推定問題を (5) 式で表される最適化問題として表現する。 $f(\mathbf{S})$ は損失関数、 $g(\mathbf{S})$ は正則化項である。

$$\min_{\mathbf{S} \in \mathbb{R}^{N \times N}} f(\mathbf{S}) + g(\mathbf{S}) \quad (5) \quad \{\text{eq:stat_op}\}$$

(4) 式のモデルに対して損失関数 $f(\mathbf{S})$ は (6) 式のように表現できる。また \mathbf{S} に関する制約を指示関数 $\iota_S(\mathbf{S})$ を用いて、(7) 式として表現する。

$$f(\mathbf{S}) := \frac{1}{2T} \|\mathbf{X} - \mathbf{S}\mathbf{X}\|_F^2 \quad (6) \quad \{\text{eq:stat_loss}\}$$

$$g(\mathbf{S}) := \iota_S(\mathbf{S}) \quad (7) \quad \{\text{eq:stat_reg}\}$$

2.3 予測-修正アルゴリズム

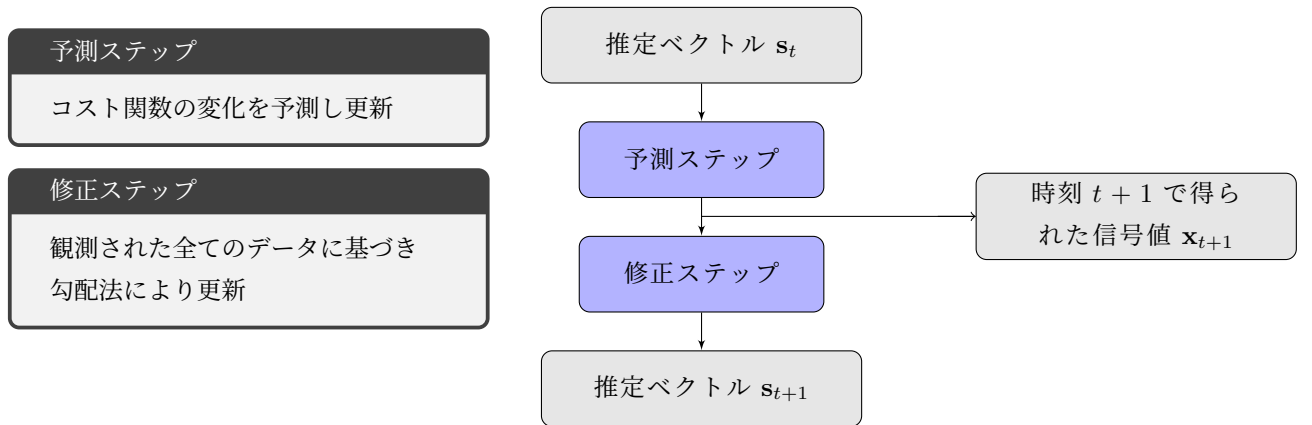


図 1: 既存法のフロー図

2.3.1 定式化

(6) 式で表される時不変な場合の SEM(構造方程式モデル: SEM) に関する損失関数 $f(\mathbf{S})$ を変形し、(9) 式のように経験的共分散行列 $\hat{\Sigma}$ で表現する。経験的共分散行列 $\hat{\Sigma}$ は (10) 式のように定義される。

$$f(\mathbf{S}) = \frac{1}{2T} \|\mathbf{X} - \mathbf{S}\mathbf{X}\|_F^2 \quad (8) \quad \{\text{stat_loss_2}\}$$

$$= \frac{1}{2} \left[\text{tr}(\mathbf{S}^2 \hat{\Sigma}) - 2\text{tr}(\mathbf{S} \hat{\Sigma}) + \text{tr}(\hat{\Sigma}) \right] \quad (9) \quad \{\text{stat_loss_3}\}$$

$$\hat{\Sigma} := \frac{1}{T} \sum_{t=1}^T \mathbf{x}_t \mathbf{x}_t^\top \quad (10) \quad \{\text{stat_sigma}\}$$

この経験的共分散行列 $\hat{\Sigma}$ を (11) 式のように再帰的に更新することで、(12) 式を時変に対応させる。また正則化の役割を果たす関数 $g(\mathbf{S}; t)$ は (13) 式のように表現する。

$$\hat{\Sigma}_t := \gamma \hat{\Sigma}_{t-1} + (1 - \gamma) \mathbf{x}_t \mathbf{x}_t^\top \quad t = 1, 2, \dots \quad (11) \quad \{\text{tv_sigma_t}\}$$

$$f(\mathbf{S}; t) := \frac{1}{2} \left[\text{tr}(\mathbf{S}^2 \hat{\Sigma}_t) - 2\text{tr}(\mathbf{S} \hat{\Sigma}_t) + \text{tr}(\hat{\Sigma}_t) \right] \quad (12) \quad \{\text{tv_loss}\}$$

$$g(\mathbf{S}; t) := \iota_S(\mathbf{S}) \quad (13) \quad \{\text{tv_reg}\}$$

ここで (14), (15) 式に示す重複行列 (Duplication matrix) \mathbf{D}_h 、除外行列 (Elimination matrix) \mathbf{E}_h を用いて、行列 \mathbf{S} をベクトル \mathbf{s} に変換する。この変換は隣接行列 \mathbf{S} に関する (3) 式の制約を用いて変数の数を削減するために行われる。また制約条件をアルゴリズム中で明示的に扱う必要がなくなる。

$$\text{vec}(\mathbf{S}) = \mathbf{D}_h \mathbf{s} \in \mathbb{R}^{N^2} \quad (14) \quad \{\text{dup_matrix}\}$$

$$\mathbf{s} = \mathbf{E}_h \text{vec}(\mathbf{S}) \in \mathbb{R}^{\frac{N(N-1)}{2}} \quad (15) \quad \{\text{eli_matrix}\}$$

以上の変数の削減により、(12) 式の損失関数 $f(\mathbf{s}; t)$ は (16) 式のように表現される。 $\mathbf{Q}_t, \hat{\sigma}_t, \hat{\sigma}_t$ はそれぞれ (17), (18), (19) 式で定義される。

$$f(\mathbf{s}; t) := \frac{1}{2} \mathbf{s}^\top \mathbf{Q}_t \mathbf{s} - 2\mathbf{s}^\top \hat{\sigma}_t + \frac{1}{2} \hat{\sigma}_t \quad (16) \quad \{\text{tv_loss2}\}$$

$$\mathbf{Q}_t := \mathbf{D}_h^\top (\hat{\Sigma}_t \otimes \mathbf{I}) \mathbf{D}_h \quad (17) \quad \{\mathbf{q}\}$$

$$\hat{\sigma}_t := \mathbf{E}_h \hat{\Sigma}_t \quad (18) \quad \{\text{bf_sigma}\}$$

$$\hat{\sigma}_t := \text{tr}(\hat{\Sigma}_t) \quad (19) \quad \{\text{sigma}\}$$

(20), (21) 式にそれぞれ予測ステップ、修正ステップの更新式を示す。ただし予測ステップでは (22) 式のように近似を行う。また (16) 式より、損失関数 $f(\mathbf{s}; t)$ の勾配 $\nabla_s f(\mathbf{s}; t)$ 、ヘシアン $\nabla_{ss} f(\mathbf{s}; t)$ 、混合 1 次偏導関数 $\nabla_{ts} f(\mathbf{s}; t)$ は (23), (24), (25) 式で表現される。 α_t, β_t はそれぞれのステップにおける固定ステップサイズである。

$$\hat{\mathbf{s}}^{p+1} = \hat{\mathbf{s}}^p - \alpha_t [\nabla_s f(\hat{\mathbf{s}}_t; t)] + \nabla_{ss} f(\hat{\mathbf{s}}_t; t) (\hat{\mathbf{s}}^p - \hat{\mathbf{s}}_t) + h \nabla_{ts} f(\hat{\mathbf{s}}_t; t) \quad (20) \quad \{\text{pred}\}$$

$$\hat{\mathbf{s}}^{c+1} = \hat{\mathbf{s}}^c - \beta_t \nabla_s f(\hat{\mathbf{s}}^c; t+1) \quad (21) \quad \{\text{corr}\}$$

$$\begin{aligned} \hat{f}(\mathbf{s}; t+1) &= \frac{1}{2} \mathbf{s}^\top \nabla_{ss} f(\hat{\mathbf{s}}_t; t) \mathbf{s} \\ &\quad + [\nabla_s f(\hat{\mathbf{s}}_t; t) + h \nabla_{ts} f(\hat{\mathbf{s}}_t; t) \mathbf{s} - \nabla_{ss} f(\hat{\mathbf{s}}_t; t) \hat{\mathbf{s}}_t]^\top \mathbf{s} \end{aligned} \quad (22) \quad \{\text{pred_aprox}\}$$

$$\nabla_s f(\mathbf{s}; t) = \mathbf{Q}_t \mathbf{s} - 2\hat{\sigma}_t \quad (23) \quad \{\text{grad}\}$$

$$\nabla_{ss} f(\mathbf{s}; t) = \mathbf{Q}_t \quad (24) \quad \{\text{hessian}\}$$

$$\nabla_{ts} f(\mathbf{s}, t) = (\mathbf{Q}_t - \mathbf{Q}_{t-1})\mathbf{s} - 2(\hat{\sigma}_t - \hat{\sigma}_{t-1}) \quad (25) \quad \{\text{mixed}\}$$

Algorithm 1 Prediction-Correction Algorithm [1]

Require: Feasible $\hat{\mathbf{S}}_0$, $f(\mathcal{S}; t_0)$, P , C , operators $\hat{\mathcal{T}}$ and \mathcal{T}

```

1:  $\hat{\mathbf{s}}_0 \leftarrow$  ad-hoc vectorization of  $\hat{\mathbf{S}}_0$ 
2: // Prediction
3: for  $t = 0, 1, \dots$  do
4:   Initialize the predicted variable as  $\hat{\mathbf{s}}^0 = \hat{\mathbf{s}}_t$ 
5:   for  $p = 0, 1, \dots, P - 1$  do
6:     Predict  $\hat{\mathbf{s}}^{p+1}$  with (15)
7:   end for
8:   Set the predicted variable  $\hat{\mathbf{s}}_{t+1|t} = \hat{\mathbf{s}}^P$ 
9:   // Correction - time  $t + 1$ : new data arrive
10:  Initialize the corrected variable as  $\hat{\mathbf{s}}^0 = \hat{\mathbf{s}}_{t+1|t}$ 
11:  for  $c = 0, 1, \dots, C - 1$  do
12:    Predict  $\hat{\mathbf{s}}^{c+1}$  with (18)
13:  end for
14:  Set the corrected variable  $\hat{\mathbf{s}}_{t+1} = \hat{\mathbf{s}}^C$ 
15: end for
```

2.4 適応並列射影劣勾配法 (Adaptive Parallel Subgradient Projection: APSP)

適応並列射影劣勾配法 (Adaptive Parallel Subgradient Projection: APSP) では、以下のように定義される閉凸集合列

$$\{C_\ell^{(k)} \subset \mathcal{H} \mid \ell \in I_k, k \in \mathbb{N}\}$$

と、対応する重み $\{w^{(\ell)} > 0 \mid \ell \in I_k, \sum_{\ell \in I_k} w^{(\ell)} = 1\}$ を用意し、それらの近似的距離関数（あるいはその上界となる劣勾配）を組み合わせることで定義される汎関数

$$\varphi_k(x) := \begin{cases} \sum_{\ell \in I_k} \frac{w^{(\ell)} d_{C_\ell^{(k)}}(h_k)}{\nu_k} d_{C_\ell^{(k)}}(x), & \text{if } \nu_k = \sum_{\ell \in I_k} w^{(\ell)} d_{C_\ell^{(k)}}(h_k) \neq 0, \\ 0, & \text{otherwise,} \end{cases}$$

により、次の反復則を構成する。ただし、 $d_{C_\ell^{(k)}}(\cdot)$ は集合 $C_\ell^{(k)}$ からの距離またはその近似的評価とみなす。

この適応並列射影劣勾配法 (Adaptive Parallel Subgradient Projection: APSP) では、各ステップでの更新規則を劣勾配法的に定義し、かつ、更新後の推定ベクトル h_{k+1} を所定の閉凸集合への射影演算 $P_K(\cdot)$ によって制限することで、次の形式の反復を与える（ μ_k はステップサイズを表す）：

$$h_{k+1} = P_K\left(h_k - \mu_k \nabla \varphi_k(h_k)\right),$$

ここで、 $\nabla \varphi_k(h_k)$ は φ_k の劣勾配（もしくはガトー微分が存在する場合はその勾配）を表す。また、適切な有界性やステップサイズ条件などを課すと、収束や収束速度に関する各種理論的性質が保証されることが示されている。

第3章 提案法

3.1 アイデア

従来法ではパラメータの設定によっては収束が遅くなる場合があるため、更新過程における収束速度を最適化するための手法が求められる。特に、予測ステップが収束速度に寄与しないような状況においては、過去のデータに依存しないアルゴリズムを用いるほうが有効な場合がある。また、グラフ構造が急激に変化するケースに対応するためには、1回の更新で使用するデータ量を適切に減らす手法が必要となる。本研究では、全てのノード間の最短経路を求める際に広く用いられる適応並列射影劣勾配法 (Adaptive Parallel Subgradient Projection: APSP) を適用しながら更新を進めることで、推定精度と更新効率のバランスを図ることを目的としている。

3.2 提案手法の優位性

本手法の大きな特徴として、ベクタライズによる変数削減を行わないために、有向グラフへも容易に対応が可能である点が挙げられる。さらに、更新方向を1つのデータに対して逐次的に計算する方式を採用しているため、複数のデータを同時に処理する並列計算を実行できる。これにより、大規模データを扱う場合においても効率よく更新を行うことが可能となり、結果として動的に変化するグラフ構造の推定に対しても高い適応能力を発揮する。

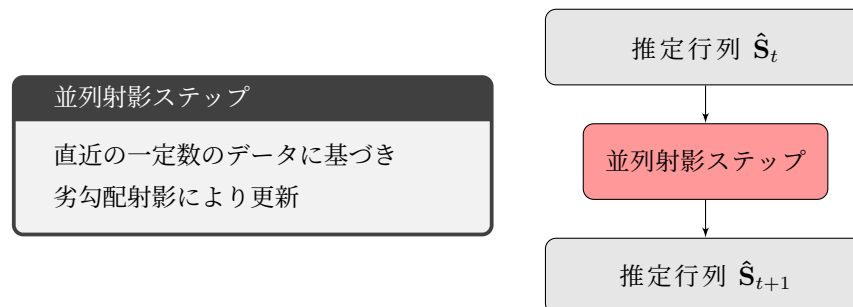


図 2: 提案手法のフロー図

3.3 提案手法の定式化

APSP の SEM(構造方程式モデル: SEM) への適用を考える。(26) 式のようにインデックス集合 \mathcal{I}_t を定義する。この集合は時刻 t における推定の更新に利用するデータのインデックスを表す。また q は並列処理を行うプロセッサの数を表す。

$$\iota \in \mathcal{I}_t := \{t, t-1, \dots, t-q+1\} \quad (26) \quad \{\text{window_set}\}$$

$r \in \mathbb{N}_+$ はプロセッサごとに使用する観測信号の数を表す。(27) 式のように r 個の観測信号をまとめて行列 $\mathbf{X}_\iota \in \mathbb{R}^{N \times r}$ で表現する。

$$\mathbf{X}_\iota := [\mathbf{x}_\iota \mathbf{x}_{\iota-1} \dots \mathbf{x}_{\iota-r+1}] \quad (27) \quad \{\mathbf{x}\}$$

プロセッサ ι における観測信号 \mathbf{X}_ι に関して、(28) 式のような集合を考える。この集合を観測信号 \mathbf{X}_ι から予測される推定行列の候補と考えられ、 ρ は候補の集合の大きさを調節するパラメータとなる。

$$C_\iota(\rho) := \left\{ \mathbf{S} \in \mathbb{R}^{N \times N} \mid \frac{1}{2} \|\mathbf{X}_\iota - \mathbf{S}\mathbf{X}_\iota\|_2^2 \leq \rho \right\} \quad (28) \quad \{\mathbf{c}\}$$

また以下に示す (29) 式のような関数 $g_\iota(\mathbf{S})$ を定義することで、集合 $C_\iota(\rho)$ を (30) 式のようにレベルセットを用いて表現できる。

$$g_\iota(\mathbf{S}) := \frac{1}{2} \|\mathbf{X}_\iota - \mathbf{S}\mathbf{X}_\iota\|_2^2 - \rho \quad (29) \quad \{\mathbf{g}\}$$

$$C_\iota(\rho) = \text{lev}_{\leq 0} g_\iota = \{ \mathbf{S} \in \mathbb{R}^{N \times N} : g_\iota(\mathbf{S}) \leq 0 \} \quad (30) \quad \{\mathbf{c_level}\}$$

(28), (29), (30) 式に基づいて、(31), (32) 式のように劣勾配射影を定めることができる。

$$\nabla g_\iota(\mathbf{S}) = \mathbf{S}\mathbf{X}_\iota\mathbf{X}_\iota^\top - \mathbf{X}_\iota\mathbf{X}_\iota^\top \quad (31) \quad \{\text{nabla_g}\}$$

$$T_{\text{sp}(g_\iota)}(\mathbf{S}) := \begin{cases} \mathbf{S} - \frac{g_\iota(\mathbf{S})}{\|\nabla g_\iota(\mathbf{S})\|^2} \nabla g_\iota(\mathbf{S}) & \text{if } g_\iota(\mathbf{S}) > 0, \\ \mathbf{S} & \text{otherwise.} \end{cases} \quad (32) \quad \{\mathbf{Tsp}\}$$

(33) 式のような並列射影を考えることができる。

$$\hat{\mathbf{S}}_{t+1} := \hat{\mathbf{S}}_t + \mu_t \left(\sum_{\iota \in \mathcal{I}_t} w_\iota^{(t)} T_{\text{sp}(g_\iota)}(\hat{\mathbf{S}}_t) - \hat{\mathbf{S}}_t \right) \quad (33) \quad \{\text{parallel}\}$$

初期推定行列 $\hat{\mathbf{S}}_0$ は、次元が $N \times N$ の実数行列として与えられる。すなわち、 $\hat{\mathbf{S}}_0 \in \mathbb{R}^{N \times N}$ である。時刻 t の推定更新において、ウィンドウに含まれるすべてのインデックス $\iota \in \mathcal{I}_t$ について、重み $w_\iota^{(t)}$ は正の実数となるように設定する。先に挙げた重み $w_\iota^{(t)}$ を、 $\iota \in \mathcal{I}_t$ のすべてにわたって合計すると 1 となるように正規化する。更新係数 (学習率) μ_t は 0 より大きく $2\mathcal{M}_t$ より小さい範囲に属する。ここで \mathcal{M}_t は (34) 式で定義される。

$$\mathcal{M}_t := \begin{cases} \frac{\sum_{\iota \in \mathcal{I}_t} w_\iota^{(t)} \|T_{\text{sp}(g_\iota)}(\hat{\mathbf{S}}_t) - \hat{\mathbf{S}}_t\|^2}{\left\| \sum_{\iota \in \mathcal{I}_t} w_\iota^{(t)} T_{\text{sp}(g_\iota)}(\hat{\mathbf{S}}_t) - \hat{\mathbf{S}}_t \right\|^2} & \text{if } \hat{\mathbf{S}}_t \notin \bigcap_{\iota \in \mathcal{I}_t} C_\iota(\rho), \\ 1 & \text{otherwise.} \end{cases} \quad (34) \quad \{\text{eq:apsp_M}\}$$

Algorithm 2 Proposed Algorithm

Require: The number of parallel processors $q \in \mathbb{N}_+$, the control sequence \mathcal{I}_t , the weights $w_\iota^{(t)}$, the input \mathbf{X}_ι , $\forall \iota \in \mathcal{I}_t, \forall t \in \mathbb{N}$, ρ , and any $\hat{\mathbf{S}}_0 \in \mathcal{S}$.

- 1: **for** $t = 0, 1, \dots$ **do**
- 2: **for all** $\iota \in \mathcal{I}_t$ **do**
- 3: $g_\iota(\hat{\mathbf{S}}_t) = \frac{1}{2} \|\mathbf{X}_\iota - \hat{\mathbf{S}}_t \mathbf{X}_\iota\|^2 - \rho$
- 4: $\nabla g_\iota(\hat{\mathbf{S}}_t) = \hat{\mathbf{S}}_t \mathbf{X}_\iota \mathbf{X}_\iota^\top - \mathbf{X}_\iota \mathbf{X}_\iota^\top$
- 5: $T_{\text{sp}(g_\iota)}(\mathbf{S}) = \begin{cases} \mathbf{S} - \frac{g_\iota(\mathbf{S})}{\|\nabla g_\iota(\mathbf{S})\|^2} \nabla g_\iota(\mathbf{S}) & \text{if } g_\iota(\mathbf{S}) > 0, \\ \mathbf{S} & \text{otherwise.} \end{cases}$
- 6: **end for**
- 7: $\mathcal{M}_t = \begin{cases} \frac{\sum_{\iota \in \mathcal{I}_t} w_\iota^{(t)} \|T_{\text{sp}(g_\iota)}(\hat{\mathbf{S}}_t) - \hat{\mathbf{S}}_t\|^2}{\left\| \sum_{\iota \in \mathcal{I}_t} w_\iota^{(t)} T_{\text{sp}(g_\iota)}(\hat{\mathbf{S}}_t) - \hat{\mathbf{S}}_t \right\|^2} & \text{if } \hat{\mathbf{S}}_t \notin \bigcap_{\iota \in \mathcal{I}_t} C_\iota(\rho), \\ 1 & \text{otherwise.} \end{cases}$
- 8: Choose $\mu_t \in (0, 2\mathcal{M}_t)$
- 9: $\hat{\mathbf{S}}_{t+1} = P_{\mathcal{S}}(\hat{\mathbf{S}}_t + \mu_t \left(\sum_{\iota \in \mathcal{I}_t} w_\iota^{(t)} T_{\text{sp}(g_\iota)}(\hat{\mathbf{S}}_t) - \hat{\mathbf{S}}_t \right))$
- 10: **end for**

第 4 章 実験設定

本節では、数値実験におけるパラメータ設定について詳細に説明する。以下に示す各パラメータは、時変構造方程式モデル (Time Varying Structural Equation Model: TV-SEM) のシミュレーションおよびオンライン推定アルゴリズムの性能評価に用いられたものである。

4.1 シミュレーションパラメータ

シミュレーションに使用した主なパラメータは以下の表 1 に示す。これらのパラメータは、グラフ構造の生成およびデータのシミュレーションにおいて重要な役割を果たす。

表 1: シミュレーションパラメータ

パラメータ	説明
N	ノード数
T	時間ステップ数
<code>max_weight</code>	隣接行列の非ゼロ要素の絶対値の最大値を 0.5 に制限
<code>variance_e</code>	ノイズの分散
K	変化点の数

{tab:simulati

4.2 初期隣接行列の生成

初期隣接行列 \mathbf{S}_0 は、対称性を持たせるために以下の手順で生成された。

1. 隣接行列の初期化： \mathbf{S}_0 はスパースなランダム行列として生成される。対称性を確保するために、生成された行列を対称行列に変換する。
2. 正規化：生成された \mathbf{S}_0 をフロベニウスノルムで正規化し、行列のノルムを 1 に調整する。

これにより、初期隣接行列 \mathbf{S}_0 は適切なスパース性と対称性を持つ形で設定される。

4.3 データ生成方法

本節では、観測データ \mathbf{X} がノイズ項 ϵ_t からどのように生成されるかを詳細に説明する。

時刻 t における観測信号 \mathbf{x}_t は、構造方程式モデル (Structural Equation Model: SEM) に基づき、以下の (35) 式で表される。

$$\mathbf{x}_t = \mathbf{S}_t \mathbf{x}_t + \epsilon_t \quad (35) \quad \{\text{eq:data_gene}\}$$

ただし、 \mathbf{S}_t は隣接行列、 ϵ_t はノイズ項であり、 $\epsilon_t \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$ と仮定する。(35) 式を整理すると、(36) 式のようになる。

$$\mathbf{x}_t = (\mathbf{I} - \mathbf{S}_t)^{-1} \epsilon_t \quad (36) \quad \{\text{eq:x_from_ep}\}$$

4.4 アルゴリズムの初期化と実行

初期隣接行列 \mathbf{S}_0 を基に、以下の 4 つのオンライン TV-SEM (Time Varying Structural Equation Model: TV-SEM) アルゴリズムを初期化し、実行した。

1. Prediction Correction (PC) アルゴリズム [1]

パラメータ $\alpha, \beta_{\text{pc}}$ を用いて、予測ステップ、修正ステップを実行するアルゴリズム。

2. Correction Only (CO) アルゴリズム [1]

パラメータ β_{pc} を用いて、修正ステップのみを実行するアルゴリズム。

3. SGD アルゴリズム [1]

パラメータ β_{sgd} を用いて、Correction Only アルゴリズムと同様に、修正ステップのみを実行するが、直近の 1 データのみを用いて更新を行う。

4. 提案手法

パラメータ r, q, ρ を用いて、並列射影ステップを組み込んだ提案手法。

各アルゴリズムは、シミュレーションデータ \mathbf{X} を入力として受け取り、時刻 t における隣接行列の推定値 $\hat{\mathbf{S}}_t$ を逐次的に更新する。推定結果および各アルゴリズムのコスト関数の値を収集し、性能評価を行った。

4.5 各アルゴリズムのパラメータ

オンライン推定アルゴリズムにおける各種パラメータは以下の表 2 に示す。これらのパラメータは、推定アルゴリズムの動作および収束特性に影響を与える重要な要素である。

表 2: オンライン TV-SEM アルゴリズムのパラメータ

パラメータ	説明
P	予測ステップの回数
C	修正ステップの回数
γ	移動平均の減衰係数
α	予測ステップにおけるステップサイズ
β	修正ステップにおけるステップサイズ
r	並列射影ステップで 1 プロセッサごとに使用するデータのウィンドウサイズ
q	並列計算を行うプロセッサの数
ρ	劣勾配射影における制約集合のサイズを調整するためのパラメータ

{tab:algorithm}

4.6 評価指標

本実験では、推定行列 $\hat{\mathbf{S}}_t$ と真の隣接行列 \mathbf{S}^* との間の正規化二乗誤差 (Normalized Squared Error: NSE) を評価指標として用いた。NSE は以下の (37) 式で定義される。

$$\text{NSE}(\hat{\mathbf{S}}_t, \mathbf{S}^*) = \frac{\|\hat{\mathbf{S}}_t - \mathbf{S}^*\|_F^2}{\|\mathbf{S}_0 - \mathbf{S}^*\|_F^2} \quad (37) \quad \{\text{eq:nse}\}$$

ここで、 $\|\cdot\|_F$ はフロベニウスノルムを表し、 \mathbf{S}^* は真の隣接行列を示す。NSE は推定誤差を初期推定誤差で正規化することで、アルゴリズムの収束性能を比較可能とする。NSE の値が小さいほど、推定行列が真の隣接行列に近いことを意味する。

4.7 実験結果

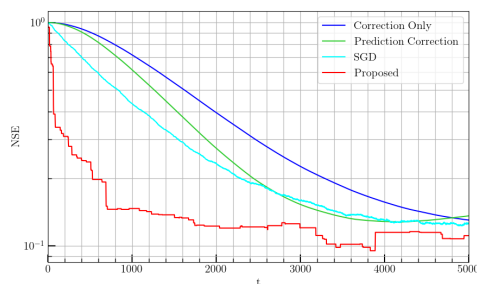
4.7.1 グラフ構造が時不変な場合

まず、グラフ構造が時間的に変化しない場合を想定し、そのときの推定性能を評価した。表 3 に、時不変のグラフ構造に対する性能評価時の各アルゴリズムの主なパラメータを示す。

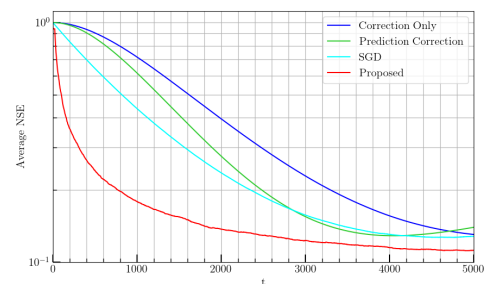
表 3: 時不変のグラフ構造に対する性能評価時のパラメータ

パラメータ	Correction Only	Prediction Correction	SGD	提案手法
P	0	1	0	-
C	1	1	1	-
γ	0.999	0.999	0	-
α	-	0.015	-	-
β	0.02	0.015	0.02	-
r	-	-	-	4
q	-	-	-	10
ρ	-	-	-	0.15

続いて、図 3 に、時不変のグラフ構造に対する各アルゴリズムの推定結果および 100 回の試行による平均を示す。図 3a では、一つの試行の結果として観測量の推定誤差の推移が示されており、図 3b では 100 回の試行について推定誤差を平均した結果を示している。提案手法が他手法と比較して良好な推定精度を得られていることが分かる。



(a) 時不変のグラフ構造に対する推定



(b) 100 回の試行による平均

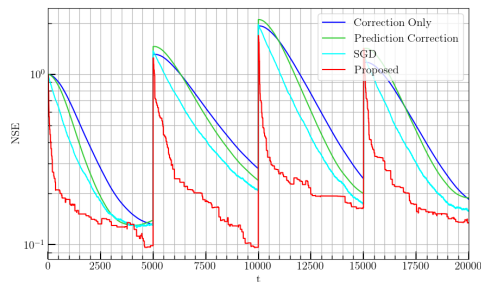
図 3: 時不変のグラフ構造に対するアルゴリズムの性能評価

4.7.2 時間方向のグラフ構造変化に対するアルゴリズムの性能評価

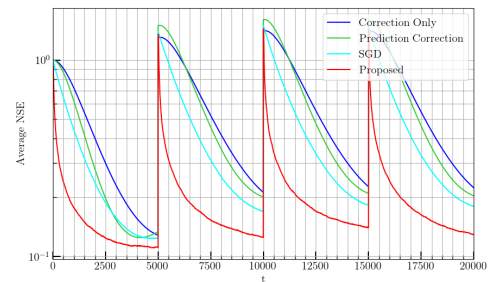
次に、時間方向においてのみグラフ構造が変化し、それ以外の箇所では一定のグラフ構造が維持されるような設定を考える。このような設定に基づいて、各アルゴリズムの性能を評価した結果を示す。まず、ノード数 10 の場合について、表 4 に主なパラメータを示し、図 4 に推定結果を示す。図 4a では、一つの試行における推定誤差の推移が、図 4b では 100 回の試行による推定誤差の平均が示されている。提案手法は他のアルゴリズムよりも高い収束性能を示しており、特に時間方向に変化が生じる局面での推定性能において差が顕著であることが分かる。

表 4: 時間方向のグラフ構造変化に対する性能評価時のパラメータ (ノード数 10)

パラメータ	Correction Only	Prediction Correction	SGD	提案手法
P	0	1	0	-
C	1	1	1	-
γ	0.999	0.999	0	-
α	-	0.015	-	-
β	0.02	0.015	0.02	-
r	-	-	-	4
q	-	-	-	10
ρ	-	-	-	0.15



(a) 時間方向のグラフ構造変化に対する推定



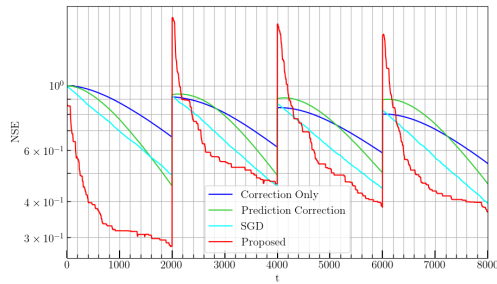
(b) 100 回の試行による平均

図 4: 時間方向のグラフ構造変化に対するアルゴリズムの性能評価 (ノード数 10)

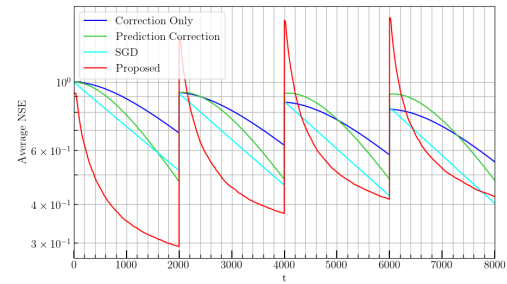
次に、ノード数 30 の場合についての結果を示す．表 5 に主なパラメータを示し，図 5 に推定結果を示す．ノード数が増加した場合でも，提案手法が他の手法に比べて安定した推定精度を維持する様子が確認できる．

表 5: 時間方向のグラフ構造変化に対する性能評価時のパラメータ (ノード数 30)

パラメータ	Correction Only	Prediction Correction	SGD	提案手法
P	0	1	0	-
C	1	1	1	-
γ	0.999	0.999	0	-
α	-	0.01	-	-
β	0.01	0.01	0.01	-
r	-	-	-	30
q	-	-	-	10
ρ	-	-	-	2.4



(a) 時間方向のグラフ構造変化に対する推定



(b) 100 回の試行による平均

図 5: 時間方向のグラフ構造変化に対するアルゴリズムの性能評価 (ノード数 30)

第 5 章 結論

本研究では、オンラインで変化するグラフ構造を推定する問題に対し、従来の予測-修正フレームワークを拡張し、適応並列射影劣勾配法 (Adaptive Parallel Subgradient Projection: APSP) を組み込んだ新たな手法を提案した。本手法により、非定常な環境下でも安定した推定性能が得られることを数値実験で示した。

特に、グラフ構造が急峻に変化する場合や、長期的に平滑な変化を示す場合など、様々な非定常性を持つシナリオにおいて、提案手法は迅速な適応と高精度な推定を両立した。また、提案法はベクタライズや制約条件への容易な対応により、計算コストおよび実装上の利点を有している。

しかし、本研究で用いたコスト関数や正則化条件は限定的であり、さらなる改良の余地がある。たとえば、他の正則化手法やより複雑な時変モデル、確率的勾配降下や変分ベイズ的アプローチなども考えられる。さらに、大規模ネットワークへの適用性や疎行列演算を活用した計算量削減も今後の課題である。

総じて、本研究の成果は、時変に変化するグラフ構造の推定問題に対して有用な一歩となり、より高度な動的ネットワーク解析手法の開発へ向けた基盤を築いたと言える。

参考文献

- [1] A. Natali, E. Isufi, M. Coutino and G. Leus, " Learning time-varying graphs from online data" , IEEE Open Journal of Signal Processing, vol. 3, pp. 212–228, 2022.
- [2] M. Yukawa, K. Slavakis, and I. Yamada, "Adaptive Parallel Quadratic-Metric Projection Algorithms," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 5, pp. 1665–1680, Jul. 2007.
- [3] I. Yamada, K. Slavakis, and K. Yamada, "An Efficient Robust Adaptive Filtering Algorithm Based on Parallel Subgradient Projection Techniques," *IEEE Transactions on Signal Processing*, vol. 50, no. 5, pp. 1091–1095, May 2002.
- [4] Y. Yamada, Adaptive Projected Subgradient Methods for Asymptotic Minimization of Sequence of Convex Functions, IEICE Transactions on Fundamentals, vol. E86-A, no. 8, pp. 1864–1869, 2003.