数理解析研究所講究録
第1999巻 2016年 154-173

確率密度微分の直接推定と機械学習への応用

奈良先端科学技術大学院大学・情報科学研究科・佐々木　博昭
Hiroaki Sasaki
Graduate School of Information Science
Nara Institute of Science and Technology
hsasaki@is.naist.jp

東京大学・大学院新領域創成科学研究科・杉山　将
Masashi Sugiyama
Graduate School of Frontier Sciences
The University of Tokyo
sugi@k.u-tokyo.ac.jp

概要

確率密度関数の微分を推定することは，機械学習の研究分野において，一般性のある研究課題である．確率密度関数の微分を推定する上で，単純なアプローチは，最初に，確率密度関数を推定し，次に，その微分を計算することであろう．しかし，この2段階推定は適切なアプローチではない．なぜならば，良い確率密度関数の推定結果が，必ずしも良い確率密度関数の微分推定結果をもたらすとは限らないからである．より適切なアプローチは，確率密度関数の推定を実行することなく，直接，確率密度関数の微分を推定することであろう．本稿の目的は，この直接推定のアプローチに沿ったいくつかの確率密度関数の微分推定法を解説することである．そして，それら微分推定法の機械学習問題への応用例についても紹介する．

1　はじめに

入力データの確率密度関数を推定することは，機械学習における問題を解く上で最も一般性のあるアプローチの1つである．その一方，確率密度関数ではなく，本質的には，そ

の微分を必要とする問題が数多く存在する．例えば，平均シフトクラスタリング [1, 2, 3] では，まず，推定された確率密度関数の勾配を用いて，データ点を確率密度関数のモード点（極大点）へ向けて更新する．そして，同じモード点に収束したデータ点に対して，同じクラスタラベルを割り当てることでクラスタリングを行う．他の例として，最近傍法を用いたカルバック・ライブラー情報量推定法の推定バイアスは，確率密度関数のヘシアン行列に依存することが知られている [4, 5]．その他にも，カーネル密度推定における最適バンド幅パラメータ決定 [6] もしくは最適バンド幅行列決定 [7, 8]，非ガウス成分分析 [9, 10]，十分次元削減 [11, 12] などの問題が確率密度関数の微分推定を用いて解くことができる．さらに，文献 [13] では，確率密度関数の微分に関連した統計データ解析問題が紹介されている．したがって，確率密度関数の微分を推定することは，機械学習研究において，一般性のある研究課題という言うことができるであろう．

確率密度関数を推定する上で，最も単純なアプローチは，(1) 入力データの確率密度関数を推定し，(2) その微分を計算するという 2 段階推定法である．しかしながら，この 2 段階推定は，確率密度関数の微分を推定する上で，適切なアプローチではない．なぜならば，良い確率密度関数の推定結果が，必ずしも良い確率密度関数の微分推定結果をもたらすとは限らないからである．加えて，このアプローチでは，高階微分を計算する際に，低次微分推定結果に基づくため，より大きな推定誤差が生じると考えられる．

この問題に対処する上で，より適切なアプローチは，確率密度関数の推定を行うことなく，直接，その微分を推定することであろう．本稿の目的は，この直接推定のアプローチに基づいた確率密度関数の微分推定法とその機械学習問題への応用結果を解説・紹介することである [14, 5, 12, 10] ．直接推定法の基本的な考え方は，2 乗損失関数の下で確率密度関数の微分モデルを真の確率密度関数の微分に直接適合することである．この単純な定式化によって，微分が解析的に推定でき，モデル選択法も組み込まれているといった利点も合わせもつ．実際に，このアプローチによって，高精度な微分推定が可能となり，既存手法を上回る性能をもつ機械学習手法がいくつか構築されている．

本稿では，第 2 節で，確率密度関数の対数勾配の直接推定法とモード探索クラスタリングへの応用結果について示す．第 3 節では，条件付き確率密度関数の対数勾配の推定法と十分次元削減への応用について紹介する．その他の微分推定法や応用例は，第 4 節で簡潔に紹介・解説する．第 5 節は，本稿のまとめである．

2 対数密度勾配推定とモード探索クラスタリングへの応用

本節では，最初に平均シフトクラスタリングについて解説し，次に，確率密度関数の対数勾配の直接推定法 [15, 14] を紹介する．そして，その直接推定法をモード探索クラスタリングへ応用し，その有用性を数値実験によって示す．

2.1 平均シフトクラスリング

平均シフトクラスタリング（mean shift clustering） [1, 2, 3] は，モード探索型のクラスタリング法であり，推定された確率密度関数のモード点に応じて，クラスタリングが実行される．平均シフトクラスタリングでは，最初に，推定された確率密度関数の勾配を用いた最急上昇法によって，データ点を近傍のモード点へと更新する．次に，同じモード点に収束したデータ点に同じクラスタラベルを割り当てる．推定された確率密度関数のモード点の数がクラスタ数に対応するため，平均シフトクラスタリングは，クラスタ数の事前入力が不要といった利点をもつ．平均シフトクラスタリングにおいて，重要なステップは，確率密度関数の勾配推定である．

確率密度関数の勾配を推定するために，平均シフトクラスタリングでは，最初に，カーネル関数 k を用いて，カーネル密度推定

$$\hat{p}_h(\boldsymbol{x}) = \frac{1}{nh^{d_{\mathrm{x}}}} \sum_{i=1}^{n} k\left(\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}{h^2}\right)$$

を実行することにより確率密度関数を推定する．次に，その勾配 $\nabla_{\boldsymbol{x}}\hat{p}_h(\boldsymbol{x})$ を計算する．この計算された確率密度関数の勾配から，データ点の更新に関して，次のように，不動点アルゴリズムを導出する．

$$\begin{aligned}
\nabla_{\boldsymbol{x}}\hat{p}_h(\boldsymbol{x}) &= \frac{2}{nh^{d_{\mathrm{x}}+2}} \sum_{i=1}^{n} [\boldsymbol{x}_i - \boldsymbol{x}] g\left(\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}{h^2}\right) \\
&= \frac{2}{nh^{d_{\mathrm{x}}+2}} \sum_{i=1}^{n} \boldsymbol{x}_i g\left(\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}{h^2}\right) - \boldsymbol{x}\frac{2}{nh^{d_{\mathrm{x}}+2}} \sum_{i=1}^{n} g\left(\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}{h^2}\right) \\
&= \frac{2}{nh^{d_{\mathrm{x}}+2}} \sum_{i=1}^{n} g\left(\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}{h^2}\right) \left[\frac{\sum_{i=1}^{n} \boldsymbol{x}_i g\left(\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}{h^2}\right)}{\sum_{i=1}^{n} g\left(\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}{h^2}\right)} - \boldsymbol{x}\right]
\end{aligned}$$

上式において $g(t) = -\frac{d}{dt}k(t)$ である．右辺の括弧内がゼロであるとき，$\nabla_{\boldsymbol{x}}\hat{p}_h(\boldsymbol{x}) = \boldsymbol{0}$ で

あるため，次のような更新式が導出される．

$$\boldsymbol{x} \leftarrow \frac{\sum_{i=1}^{n} \boldsymbol{x}_i g\left(\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}{h^2}\right)}{\sum_{i=1}^{n} g\left(\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}{h^2}\right)} \tag{1}$$

単純な計算により，更新式 (1) は，$\boldsymbol{x}$ に依存したステップ幅をもつ最急上昇法と等価であることを確認できる．

平均シフトクラスタリングはクラスタ数の事前設定が不要という利点をもつが，その一方，入力データの次元が高いときに，クラスタリング性能精度が低くなることが知られている [3]．1 つの理由として挙げられるのは，確率密度関数の勾配推定における 2 段階推定である．なぜならば，良い確率密度関数の推定結果が必ずしも良い確率密度関数の勾配推定結果をもたらすとは限らないからである．より適切なアプローチは，確率密度関数の推定を行うことなく，直接，確率密度関数の勾配を推定することであろう．以下では，この直接推定のアプローチに沿う確率密度関数の対数勾配の推定法 [15, 14] を紹介する．

2.2 最小 2 乗対数密度勾配

最初に，確率密度関数の対数勾配（以下，対数密度勾配と呼ぶ）推定の問題設定について述べる．確率密度関数 $p(\boldsymbol{x})$ より生成された n 個のデータ点 $\{\boldsymbol{x}_i = (x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(d_{\mathrm{x}})})^\top\}_{i=1}^n$ が与えられているとしよう．ただし，$^\top$ は転置を表す．ここでの目的は，$\{\boldsymbol{x}_i\}_{i=1}^n$ から

$$\partial_j \log p(\boldsymbol{x}) = \frac{\partial_j p(\boldsymbol{x})}{p(\boldsymbol{x})}$$

を推定することである．上式において，$\partial_j = \frac{\partial}{\partial x^{(j)}}$ である．

基本的なアプローチは，勾配モデル $\boldsymbol{r}(\boldsymbol{x}) = (r^{(1)}(\boldsymbol{x}), r^{(2)}(\boldsymbol{x}), \ldots, r^{(d_{\mathrm{x}})}(\boldsymbol{x}))^\top$ を真の対数密度勾配に対して，2 乗損失関数下で，適合することである．

$$J(r^{(j)}) = \int \left\{ r^{(j)}(\boldsymbol{x}) - \frac{\partial_j p(\boldsymbol{x})}{p(\boldsymbol{x})} \right\}^2 p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \tag{2}$$

式 (2) を展開することで 3 つの項を得る．

$$J(r^{(j)}) = \int \left\{ r^{(j)}(\boldsymbol{x}) \right\}^2 p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} - 2 \int r^{(j)}(\boldsymbol{x}) \partial_j p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} + \int \left\{ \frac{\partial_j p(\boldsymbol{x})}{p(\boldsymbol{x})} \right\}^2 p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \tag{3}$$

式 (3) における第 1 項はデータ点より推定可能，第 3 項はモデル $r^{(j)}$ に依存しないため省略可能である．しかし，第 2 項は真の確率密度関数の微分 $\partial_j p(\boldsymbol{x})$ を含んでいるため，

推定は一見容易ではない．しかしながら，次のように部分積分を実行することで容易に推定可能な形式へと変換することができる．

$$\int r^{(j)}(\boldsymbol{x})\partial_j p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = \int \left[r^{(j)}(\boldsymbol{x})p(\boldsymbol{x})\right]_{x^{(j)}=-\infty}^{x^{(j)}=\infty} \mathrm{d}\boldsymbol{x}_{\backslash x^{(j)}} - \int \left\{\partial_j r^{(j)}(\boldsymbol{x})\right\} p(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$$
$$= -\int \left\{\partial_j r^{(j)}(\boldsymbol{x})\right\} p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \tag{4}$$

式 (4) における $\mathrm{d}\boldsymbol{x}_{\backslash x^{(j)}}$ は，$x^{(j)}$ を除いた変数についての積分を意味し，さらに，$\lim_{|x^{(j)}|\to\infty} r^{(j)}(\boldsymbol{x})p(\boldsymbol{x}) = 0$ を仮定している．同様の操作は，これまでもスタインの補題 [16] やスコアマッチング法 [17] を導くためにも用いられている．式 (3) と (4) より，次のように経験損失関数を導出することができる．

$$\widehat{J}(r^{(j)}) = \frac{1}{n}\sum_{i=1}^{n}\left\{r^{(j)}(\boldsymbol{x}_i)\right\}^2 + 2\partial_j r^{(j)}(\boldsymbol{x}_i) \tag{5}$$

ただし，式 (5) において，式 (3) の第 3 項は省略されている．

$r^{(j)}$ を推定するために，次のような線形モデルを考える．

$$r^{(j)}(\boldsymbol{x}) = \sum_{i=1}^{b}\theta_{ij}\psi_{ij}(\boldsymbol{x}) = \boldsymbol{\theta}_j^\top \boldsymbol{\psi}_j(\boldsymbol{x}) \tag{6}$$

式 (6) 内の θ_{ij} は推定されるパラメータ，$\psi_{ij}(\boldsymbol{x})$ は基底関数，b は基底関数の数を示し，本稿では $b = \min(100, n)$ に固定することにする．式 (6) を経験損失関数 (5) に代入し，ℓ_2 正則化項を加えた後で，$\boldsymbol{\theta}_j$ に関して，次のように解析解を計算できる．

$$\widehat{\boldsymbol{\theta}}_j = \operatorname*{argmin}_{\boldsymbol{\theta}_j}\left[\boldsymbol{\theta}_j^\top \mathbf{G}_j\boldsymbol{\theta}_j + 2\boldsymbol{\theta}_j^\top \boldsymbol{h}_j + \lambda_j\boldsymbol{\theta}_j^\top\boldsymbol{\theta}_j\right] = -\left(\mathbf{G}_j + \lambda_j\mathbf{I}_{d_x}\right)^{-1}\boldsymbol{h}_j$$

上式における $\mathbf{I}_{d_x}$ は d_x 行 d_x 列の単位行列，λ_j は非負の正則化パラメータ，

$$\mathbf{G}_j = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\psi}_j(\boldsymbol{x}_i)\boldsymbol{\psi}_j(\boldsymbol{x}_i)^\top, \qquad \boldsymbol{h}_j = \frac{1}{n}\sum_{i=1}^{n}\partial_j\boldsymbol{\psi}_j(\boldsymbol{x}_i)$$

最終的に，対数密度勾配の推定結果は次のような形式で与えられる．

$$\widehat{\boldsymbol{r}}(\boldsymbol{x}) = (\widehat{r}^{(1)}(\boldsymbol{x}), \widehat{r}^{(2)}(\boldsymbol{x}), \ldots, \widehat{r}^{(d_x)}(\boldsymbol{x}))^\top = (\widehat{\boldsymbol{\theta}}_1^\top\boldsymbol{\psi}_1(\boldsymbol{x}), \widehat{\boldsymbol{\theta}}_2^\top\boldsymbol{\psi}_2(\boldsymbol{x}), \ldots, \widehat{\boldsymbol{\theta}}_{d_x}^\top\boldsymbol{\psi}_{d_x}(\boldsymbol{x}))^\top$$

この手法を**最小 2 乗対数密度勾配** (least-squares log-density gradient: LSLDG) と呼ぶ．

2.3 モード探索クラスタリングへの応用

ここでは，LSLDG をモード探索クラスタリングへ応用する．平均シフトクラスタリングと同様の不動点法に基づき，クラスタリングアルゴリズムを導出する．そのアルゴリズムを導出する上で，次のような基底関数を用いる．

$$\psi_{ij}(\boldsymbol{x}) = \partial_j \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{c}_i\|^2}{2\sigma^2}\right) = \frac{c_i^{(j)} - x^{(j)}}{\sigma^2} \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{c}_i\|^2}{2\sigma^2}\right) = \frac{c_i^{(j)} - x^{(j)}}{\sigma^2}\phi_i(\boldsymbol{x})$$

上式において，$\sigma(>0)$ はバンド幅パラメータ，$\boldsymbol{c}_i = (c_i^{(1)}, c_i^{(2)}, \ldots, c_i^{(d_x)})^\top$ はガウス関数の中心点であり，データ点 $\{\boldsymbol{x}_i\}_{i=1}^b$ からランダムに b 個選択することにする．この基底関数を用いて，

$$\begin{aligned}\widehat{r}^{(j)}(\boldsymbol{x}) = \sum_{i=1}^b \widehat{\theta}_{ij}\psi_i(\boldsymbol{x}) &= \frac{1}{\sigma^2}\left[\sum_{i=1}^b \widehat{\theta}_{ij}c_i^{(j)}\phi_i(\boldsymbol{x}) - x^{(j)}\sum_{i=1}^b \widehat{\theta}_{ij}\phi_i(\boldsymbol{x})\right] \\ &= \frac{\sum_{i=1}^b \widehat{\theta}_{ij}\phi_i(\boldsymbol{x})}{\sigma^2}\left[\frac{\sum_{i=1}^b \widehat{\theta}_{ij}c_i^{(j)}\phi_i(\boldsymbol{x})}{\sum_{i=1}^b \widehat{\theta}_{ij}\phi_i(\boldsymbol{x})} - x^{(j)}\right] \end{aligned} \tag{7}$$

式 (7) より，右辺の括弧内がゼロのとき，$r^{(j)}(\boldsymbol{x}) = 0$ となる．これにより，次のような更新式を得る．

$$x^{(j)} \leftarrow \frac{\sum_{i=1}^b \widehat{\theta}_{ij}c_i^{(j)}\phi_i(\boldsymbol{x})}{\sum_{i=1}^b \widehat{\theta}_{ij}\phi_i(\boldsymbol{x})} \tag{8}$$

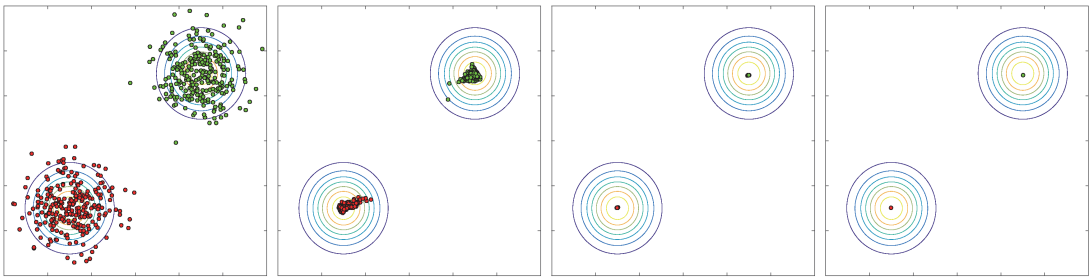
本稿では，この手法を**最小二乗対数密度勾配クラスタリング** (LSLDG clustering: LSLDGC) と呼ぶ．平均シフトクラスタリングの更新式 (1) と比較すると，最大の違いは，重み $\widehat{\theta}_{ij}$ である．したがって，LSLDGC は重み付き平均シフトクラスタリングと解釈できる．

2.4 数値実験

ここで，LSLDGC の性能を簡単な数値実験で示す．本数値実験において，入力データは，次のような混合ガウス分布より生成する．

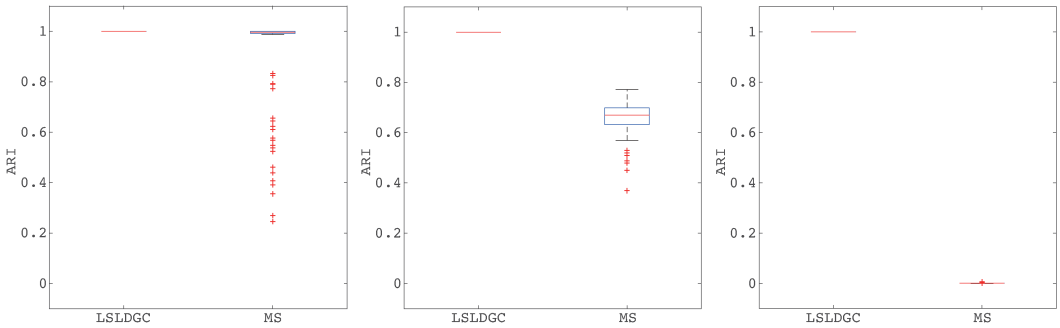
$$p(\boldsymbol{x}) = \frac{1}{2}\mathcal{N}(\boldsymbol{\mu}_1, \mathbf{I}_{d_x}) + \frac{1}{2}\mathcal{N}(\boldsymbol{\mu}_2, \mathbf{I}_{d_x})$$

$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{C})$ は期待値 $\boldsymbol{\mu}$ と共分散行列 $\boldsymbol{C}$ をもつガウス分布，$\boldsymbol{\mu}_1 = (3, 3, 0, \ldots, 0)^\top$，$\boldsymbol{\mu}_2 = (-3, -3, 0, \ldots, 0)^\top$ とした．LSLDG のパラメータ σ と λ_j は損失関数 J を評価基準と



(a) データ更新前　(b) 1 更新後　(c) 3 更新後　(d) 9 更新後

図 1　LSLDGC によるモード探索過程におけるデータ点の分布.



(a) $d_{\mathrm{x}} = 4$　(b) $d_{\mathrm{x}} = 8$　(c) $d_{\mathrm{x}} = 12$

図 2　LSLDGC と平均シフトクラスタリング（MS）の比較．箱ひげ図は，100 試行の数値実験結果である．ただし，$n = 500$.

して，交差検定によって決定した．また，カーネル関数を $k(t) = \exp(-t)$ とし，平均シフトクラスタリングを同じデータに適用し，LSLDGC に対する比較を行った．平均シフトクラスタリングにおけるバンド幅パラメータ h もまた交差検定により決定した．クラスタリング性能を評価するために，調整ランド指数（adjusted Rand index: ARI）[18] を用いた．ARI は 1 以下の値をとり，大きな ARI 値は良いクラスタリング結果を意味する．

更新式 (8) によるデータ点の更新過程を図 1 に示す．最初の数更新で，急速にデータ点が近傍のモードに向かい，その後，各モード点にデータ点が収束していく様子が分かる．このデータ点が更新されていく様子は，平均シフトクラスタリングと定性的に同じである．しかしながら，入力データの次元が高くなると，平均シフトクラスタリングの ARI

値は急速に減少していくことが分かる（図 2）．これに対して，LSLDGC はデータの次元が上がっても，高い ARI 値を維持し続ける．これらの結果は，LSLDGC が，より高次元のデータに対して，高精度なクラスタリングを実現することを意味している．

3 条件付き対数密度勾配推定と十分次元削減への応用

本節では，最初に，十分次元削減における条件付き確率密度関数の勾配に基づいたアプローチについて解説し，次に，[12] で提案された条件付き確率密度関数の対数密度勾配推定法とその十分次元削減への応用例について紹介する．

3.1 十分次元削減と条件付き確率密度関数の勾配に基づいたアプローチ

十分次元削減 (sufficient dimension reduction) [11, 19, 20] とは，教師付き次元削減の枠組みであり，出力と入力に関するデータの組を $(y, \boldsymbol{x})$ で表すと，次のような関係を満たす変換行列 $\boldsymbol{B} \in \mathbb{R}^{d_{\mathrm{x}} \times d_{\mathrm{z}}}$ を推定することを目的とする．

$$p(y|\boldsymbol{x}) = p(y|\boldsymbol{B}^{\top}\boldsymbol{x}) \tag{9}$$

$p(y|\boldsymbol{x})$ は入力データ $\boldsymbol{x}$ が与えられたときの出力データ y の条件付き確率であり，$1 \leq d_{\mathrm{z}} < d_{\mathrm{x}}$，$\boldsymbol{B}^{\top}\boldsymbol{B} = \mathbf{I}_{d_{\mathrm{z}}}$ とする．条件式 (9) は，次元削減後の入力データ $\boldsymbol{B}^{\top}\boldsymbol{x}$ は，出力データ y について，次元削減前の入力データ $\boldsymbol{x}$ と同じ情報をもつことを意味する．したがって，なるべく条件式 (9) を満たすような変換行列 $\boldsymbol{B}$ を推定することで，出力データ y の情報を失うことなく，入力データ $\boldsymbol{x}$ の次元を削減できることが期待できる．

これまでいくつかの十分次元削減法が提案されている．初期の手法は，スライス逆回帰 (sliced inverse regression) と呼ばれる手法である [11]．しかしながら，スライス逆回帰は，$\boldsymbol{x}$ の周辺密度関数 $p(\boldsymbol{x})$ が楕円型であることを仮定しているため，確率密度関数に関して，強い制約をもつ．これに対して，近年，確率密度関数に関して，制約の少ない手法が提案されている．例えば，カーネル次元削減 (kernel dimension reduction) [21] や最小 2 乗次元削減 (least-squares dimensionality reduction) [22] などがある．しかし，これら手法は，$\boldsymbol{B}$ を推定する上で，非凸最適化問題を解く必要があり，一般的には局所最適解しか得られない．

この局所最適解を避けるアプローチとして，条件付き確率密度関数の勾配に基づいたアプローチがある [23]．条件式 (9) の両辺の $\boldsymbol{x}$ に関する勾配 $\nabla_{\boldsymbol{x}}$ を計算することで，

$$\nabla_{\boldsymbol{x}} p(y|\boldsymbol{x}) = \boldsymbol{B}\nabla_{\boldsymbol{B}^{\top}\boldsymbol{x}} p(y|\boldsymbol{B}^{\top}\boldsymbol{x}) \tag{10}$$

が得られる．式 (10) より，条件付き確率密度関数の勾配 $\nabla_{\boldsymbol{x}} p(y|\boldsymbol{x})$ は，$\boldsymbol{B}$ の列空間に存在することが分かる．したがって，主成分分析と同様に，$\boldsymbol{B}$ の列ベクトルは，

$$E\{(\nabla_{\boldsymbol{x}} p(y|\boldsymbol{x}))(\nabla_{\boldsymbol{x}} p(y|\boldsymbol{x}))^{\top}\}$$

の上位 d_z 個の固有値に対応した固有ベクトルとして得られる．このアプローチは，固有値分解に基づくため，大域最適解が保障される．ただし，このアプローチにおける困難な点は，$\nabla_{\boldsymbol{x}} p(y|\boldsymbol{x})$ の推定である．

先行研究 [23] では，$\nabla_{\boldsymbol{x}} p(y|\boldsymbol{x})$ を局所線形平滑化（local linear smoother）[24] と呼ばれる手法で推定している．しかしながら，局所線形平滑化は，1 次のテイラー近似に基づいており，データ点が疎になる高次元データに対して，推定精度が落ちることが考えられる．また，パラメータ調整が困難なことや，データ点数が大きくなると計算効率が悪いといった問題点もある．そこで，[12] では，条件付き確率密度関数の対数勾配について，テイラー近似を使用することなく，モデル選択方法も含む推定手法を提案し，新たな十分次元削減法を構築している．以下では，その手法について解説する．

3.2 最小 2 乗条件付き対数密度勾配と十分次元削減への応用

最初に，条件付き確率密度関数の対数勾配（以下，条件付き対数密度勾配と呼ぶ）推定の問題設定について述べる．同時確率密度関数 $p(y,\boldsymbol{x})$ より生成された n 個の出入力データ点の組 $\{(y_i,\boldsymbol{x}_i)\}_{i=1}^{n}$ が与えられているとしよう．ここでの目的は，$\{(y_i,\boldsymbol{x}_i)\}_{i=1}^{n}$ から

$$\partial_j \log p(y|\boldsymbol{x}) = \frac{\partial_j p(y|\boldsymbol{x})}{p(y|\boldsymbol{x})} = \frac{\partial_j p(y,\boldsymbol{x})p(\boldsymbol{x}) - p(y,\boldsymbol{x})\partial_j p(\boldsymbol{x})}{p(y,\boldsymbol{x})p(\boldsymbol{x})} = \frac{\partial_j p(y,\boldsymbol{x})}{p(y,\boldsymbol{x})} - \frac{\partial_j p(\boldsymbol{x})}{p(\boldsymbol{x})}$$

を推定することである．

LSLDG と同様に，基本的なアプローチは，勾配モデル $\boldsymbol{g}(y,\boldsymbol{x}) = (g^{(1)}(y,\boldsymbol{x}),\ldots,g^{(d_{\mathrm{x}})}(y,\boldsymbol{x}))^{\top}$ を 2 乗損失関数の下で，真の条件付き対数密度勾配に直

接適合することである．

$$
\begin{aligned}
J(g^{(j)}) &= \iint \left\{ g^{(j)}(y,\boldsymbol{x}) - \partial_j \log p(y|\boldsymbol{x}) \right\}^2 p(y,\boldsymbol{x}) \mathrm{d}y\mathrm{d}\boldsymbol{x} \\
&= \iint \left\{ g^{(j)}(y,\boldsymbol{x}) \right\}^2 p(y,\boldsymbol{x}) \mathrm{d}y\mathrm{d}\boldsymbol{x} - 2 \iint g^{(j)}(y,\boldsymbol{x}) \partial_j p(y,\boldsymbol{x}) \mathrm{d}y\mathrm{d}\boldsymbol{x} \\
&+ 2 \iint g^{(j)}(y,\boldsymbol{x}) \left\{ \frac{\partial_j p(\boldsymbol{x})}{p(\boldsymbol{x})} \right\} p(y,\boldsymbol{x}) \mathrm{d}y\mathrm{d}\boldsymbol{x} + \iint \left\{ \partial_j \log p(y|\boldsymbol{x}) \right\}^2 p(y,\boldsymbol{x}) \mathrm{d}y\mathrm{d}\boldsymbol{x} \\
&= \iint \left\{ g^{(j)}(y,\boldsymbol{x}) \right\}^2 p(y,\boldsymbol{x}) \mathrm{d}y\mathrm{d}\boldsymbol{x} + 2 \iint \partial_j g^{(j)}(y,\boldsymbol{x}) p(y,\boldsymbol{x}) \mathrm{d}y\mathrm{d}\boldsymbol{x} \\
&+ 2 \iint g^{(j)}(y,\boldsymbol{x}) \left\{ \frac{\partial_j p(\boldsymbol{x})}{p(\boldsymbol{x})} \right\} p(y,\boldsymbol{x}) \mathrm{d}y\mathrm{d}\boldsymbol{x} + \iint \left\{ \partial_j \log p(y|\boldsymbol{x}) \right\}^2 p(y,\boldsymbol{x}) \mathrm{d}y\mathrm{d}\boldsymbol{x}
\end{aligned}
\tag{11}
$$

最後の等式は，LSLDG と同様に，右辺第 2 項に対して部分積分を実行した．式 (11) の右辺における第 4 項は省略可能，第 1 項と第 2 項はデータ点から容易に推定可能であるが，第 3 項を推定することが容易ではない．なぜならば，被積分関数に，真の確率密度関数の対数微分 $\partial_j \log p(\boldsymbol{x}) = \frac{\partial_j p(\boldsymbol{x})}{p(\boldsymbol{x})}$ が含まれているからである．この問題に対処するために，ここでは，真の確率密度関数の対数微分 $\partial_j \log p(\boldsymbol{x})$ を第 2 節で紹介した LSLDG による推定結果 $\widehat{r}^{(j)}(\boldsymbol{x})$ で置き換える．これにより，式 (11) の第 3 項は

$$
\iint g^{(j)}(y,\boldsymbol{x}) \left\{ \frac{\partial_j p(\boldsymbol{x})}{p(\boldsymbol{x})} \right\} p(y,\boldsymbol{x}) \mathrm{d}y\mathrm{d}\boldsymbol{x} \approx \iint g^{(j)}(y,\boldsymbol{x}) \widehat{r}^{(j)}(\boldsymbol{x}) p(y,\boldsymbol{x}) \mathrm{d}y\mathrm{d}\boldsymbol{x}
$$

と近似されるため，結果として，右辺はデータ点より容易に推定可能である．以上より，LSLDG を用いた近似経験損失関数は，

$$
\widehat{J}(g^{(j)}) = \frac{1}{n} \sum_{i=1}^{n} \left\{ g^{(j)}(y_i,\boldsymbol{x}_i) \right\}^2 + 2 \left\{ \partial_j g^{(j)}(y_i,\boldsymbol{x}_i) + g^{(j)}(y_i,\boldsymbol{x}_i) \widehat{r}^{(j)}(\boldsymbol{x}_i) \right\} \tag{12}
$$

となる．

次に，LSLDG と同様に，次のような線形モデルを用いる．

$$
g^{(j)}(y,\boldsymbol{x}) = \sum_{i=1}^{b} \theta_{ij} \underbrace{\exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|^2}{2(\sigma_x^{(j)})^2} - \frac{(y-y_i)^2}{2\sigma_y^2} \right)}_{\varphi_{ij}(y,\boldsymbol{x})} = \boldsymbol{\theta}_j^\top \boldsymbol{\varphi}_j(y,\boldsymbol{x})
$$

$\sigma_{\mathrm{x}}^{(j)}$ と σ_{y} はバンド幅パラメータである．線形モデルと ℓ_2 正則化項を近似経験損失関数 (12) に挿入することで，経験近似損失関数は 2 次形式となり，解析的に解を計算で

入力： 出入力データ $\{(y_i, \boldsymbol{x}_i)\}_{i=1}^n$.

ステップ 1 LSLCG を用いて，条件付き対数密度勾配 $\nabla_{\boldsymbol{x}} \log p(y|\boldsymbol{x})$ を推定.
ステップ 2 推定された勾配 $\widehat{\boldsymbol{g}}(y, \boldsymbol{x})$ から，$\widehat{\boldsymbol{\Lambda}} = \frac{1}{n}\sum_{i=1}^n \widehat{\boldsymbol{g}}(y_i, \boldsymbol{x}_i)\widehat{\boldsymbol{g}}(y_i, \boldsymbol{x}_i)^\top$ を計算.
ステップ 3 $\widehat{\boldsymbol{\Lambda}}$ に対して固有値分解を実行．上位 d_z 個の固有値に対応する固有ベクトルを $\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_{d_z}$ とする.

出力：$\widehat{\boldsymbol{B}} = (\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_{d_z})$.

図 3 LSGDR アルゴリズム.

きる.

$$\widehat{\boldsymbol{\theta}}_j = \arg\min_{\boldsymbol{\theta}_j} \left[\boldsymbol{\theta}_j^\top \mathbf{G}_j \boldsymbol{\theta}_j + 2\boldsymbol{\theta}_j^\top \boldsymbol{h}_j + \lambda_j \boldsymbol{\theta}_j^\top \boldsymbol{\theta}_j\right] = -(\mathbf{G}_j + \lambda_j \mathbf{I})^{-1} \boldsymbol{h}_j$$

上の式において，

$$\mathbf{G}_j = \frac{1}{n}\sum_{i=1}^n \boldsymbol{\varphi}_j(y_i, \boldsymbol{x}_i)\boldsymbol{\varphi}_j(y_i, \boldsymbol{x}_i)^\top, \qquad \boldsymbol{h}_j = \frac{1}{n}\sum_{i=1}^n \partial_j \boldsymbol{\varphi}_j(y_i, \boldsymbol{x}_i) + \widehat{r}^{(j)}(\boldsymbol{x}_i)\boldsymbol{\varphi}_j(y_i, \boldsymbol{x}_i)$$

最終的に，条件付き対数密度勾配の推定結果は，

$$\widehat{\boldsymbol{g}}(y, \boldsymbol{x}) = (\widehat{\boldsymbol{\theta}}_1^\top \boldsymbol{\varphi}_1(y, \boldsymbol{x}), \widehat{\boldsymbol{\theta}}_2^\top \boldsymbol{\varphi}_2(y, \boldsymbol{x}), \ldots, \widehat{\boldsymbol{\theta}}_{d_x}^\top \boldsymbol{\varphi}_{d_x}(y, \boldsymbol{x}))^\top$$

で与えられる．この手法を**最小 2 乗条件付き対数密度勾配** (least-squares logarithmic conditional density gradients: LSLCG) と呼ぶ.

LSLCG を応用した十分次元削減法のアルゴリズムを図 3 に示す．この十分次元削減法を**最小 2 乗勾配次元削減** (least-squares gradients for dimension reduction: LSGDR) と呼ぶ.

3.3 数値実験

ここで，LSGDR の推定性能に関する数値実験を行う．比較対象として，カーネル次元削減（KDR） [21] *1 と最小 2 乗次元削減（LSDR） [22] *2 に加え，dOPG [23]*3 と呼

*1 http://www.ism.ac.jp/~fukumizu/software.html
*2 http://www.ms.k.u-tokyo.ac.jp/software.html#LSDR
*3 http://www.stat.nus.edu.sg/~staxyc/dOPG.m

ばれる条件付き確率密度関数の勾配に基づく手法を用いた．本数値実験において，出力データ y は，モデル

$$y = \sum_{i=1}^{d_z} |x^{(i)}| + 0.3\epsilon$$

によって生成した．ただし，$x^{(i)}$ と ϵ は標準正規分布に従う確率変数である．このとき，d_x 行 d_z 列の零行列を $O_{d_x \times d_z}$ と表すと，最適な変換行列 $\boldsymbol{B}$ は，

$$\boldsymbol{B} = \begin{pmatrix} \mathbf{I}_{d_z} \\ O_{(d_x - d_z) \times d_z} \end{pmatrix}$$

となる．LSLCG におけるパラメータ $\sigma_x^{(j)}$ と λ_j は交差検定により決定，σ_y は $|y_i - y_j|$ の i, j に関する中央値とした．推定誤差は，$\|\boldsymbol{B}\boldsymbol{B}^\top - \widehat{\boldsymbol{B}}\widehat{\boldsymbol{B}}^\top\|_{\mathrm{F}}$ によって評価した．ただし，$\widehat{\boldsymbol{B}}$ は各手法の推定結果，$\|\cdot\|_{\mathrm{F}}$ はフロベニウスノルムを表す．

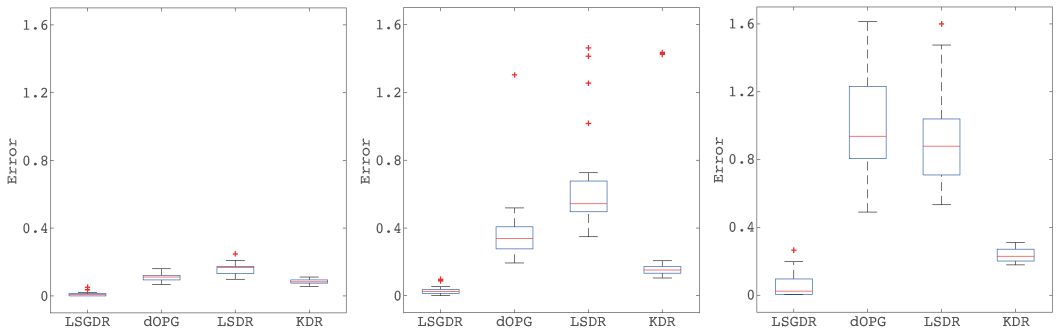
部分空間の次元 $d_z = 2, 4, 6$ に対して，LSGDR と KDR の推定誤差が小さいことが分かる（図 4）．一方，dOPG と LSDR の推定誤差は，部分空間の次元が上がるにつれて，急激に大きくなることが分かる．LSDR は，部分空間の次元が上がるほど，$\boldsymbol{B}$ の初期値の選択が難しく，望ましくない局所最適解が得られている可能性がある．dOPG は，部分空間の次元が高いとき，アルゴリズム内で用いている局所線形平滑化の精度が悪いことが，誤差の増大の原因であると考えられる．KDR の推定誤差は小さいものの，データ点数 n が大きくなるにつれて，計算時間が非常に大きくなることが分かる（図 5）．これは，LSGDR や dOPG とは異なり，KDR が反復最適化を実行するためである．dOPG の局所線形平滑化では，n^2 個のパラメータを推定する必要があり，データ点数が大きくなると計算時間が LSGDR よりも大きくなる（図 5(b,c)）．以上より，推定精度と計算時間の両面において，LSGDR は良い性能を示す手法であることが分かる．

4 確率密度微分に関連した他のトピック

本節では，他の確率密度関数の微分推定法や応用結果について紹介する．

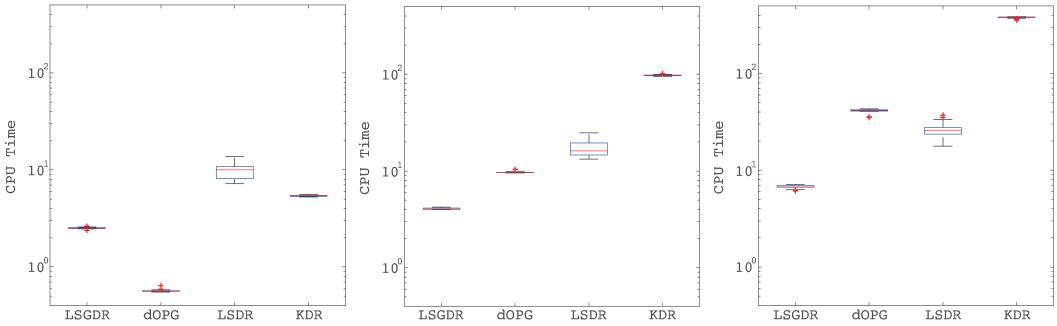
4.1 確率密度関数の高階微分の直接推定法とその応用

LSLDG は確率密度関数の対数の 1 階微分を直接推定する手法であったが，この直接推定法は，確率密度関数の任意の階数の微分を直接推定する手法へと一般化できる [5, 8].



(a) $d_z = 2$ (b) $d_z = 4$ (c) $d_z = 6$

図 4　部分空間の次元 d_z に対する各手法の推定誤差．ただし，$(d_x, n) = (10, 500)$ である．



(a) $n = 200$ (b) $n = 600$ (c) $n = 1000$

図 5　データ点数 n に対する各手法の計算時間．縦軸は，対数目盛で表示された CPU 時間である．ただし，$(d_x, d_z) = (10, 2)$ である．

例えば，確率密度関数の k 階微分を

$$p_{k,j}(\boldsymbol{x}) = \frac{\partial^k}{\partial (x^{(1)})^{j_1} \partial (x^{(2)})^{j_2} \dots \partial (x^{(d)})^{j_d}} p(\boldsymbol{x})$$

と定義し，次の 2 乗損失関数を k 階微分モデル $g_{k,j}$ に関して最小化することで推定できる．

$$J_{k,j}(g_{k,j}) = \int \{g_{k,j}(x) - p_{k,j}(\boldsymbol{x})\}^2 \, \mathrm{d}\boldsymbol{x} \tag{13}$$

ただし，$\boldsymbol{j} = (j_1, j_2, \dots, j_d)^\top$，$j_i \in \{0, 1, \dots, d\}$，$j_1 + j_2 + j_d = k$ である．式 (13) は，真の確率密度関数の k 階を含んでいるため，一見，取り扱いが困難なように見えるが，

LSLDG における部分積分の操作を k 回繰り返すことで，次のような経験損失関数を導出できる．

$$\widehat{J}_{k,j}(g_{k,j}) = \int g_{k,j}(\boldsymbol{x})^2 \mathrm{d}\boldsymbol{x} - \frac{2(-1)^k}{n} \sum_{i=1}^{n} \frac{\partial^k}{\partial (x^{(1)})^{j_1} \partial (x^{(2)})^{j_2} \ldots \partial (x^{(d)})^{j_d}} g_{k,j}(\boldsymbol{x}_i) \quad (14)$$

LSLDG と LSLCG と同様に，$g_{k,j}(\boldsymbol{x})$ に線形モデルを導入することで，解析的に微分を推定できる．この手法は，**積分 2 乗誤差密度微分推定法** (integrated squared error for density derivatives: ISED) と呼ばれている．

次に，確率密度関数の 2 階微分が必要な問題として，カルバック・ライブラー情報量推定法のバイアス削減を紹介する．確率密度関数 $p_1(\boldsymbol{x})$ と確率密度関数 $p_2(\boldsymbol{x})$ のカルバック・ライブラー情報量は，

$$\mathrm{KL}(p_1 \| p_2) = \int p_1(\boldsymbol{x}) \log \frac{p_1(\boldsymbol{x})}{p_2(\boldsymbol{x})} \mathrm{d}\boldsymbol{x}$$

で定義される．実用的なカルバック・ライブラー情報量推定法の 1 つとして，最近傍法を活用した推定法がある [25]．$p_1(\boldsymbol{x})$ と $p_2(\boldsymbol{x})$ から生成されたデータ点をそれぞれ $\mathcal{X}_1 = \{\boldsymbol{x}_i\}_{i=1}^{n_1}$，$\mathcal{X}_2 = \{\boldsymbol{x}_i\}_{i=n_1+1}^{n_1+n_2}$ とすると，

$$\widehat{\mathrm{KL}}(p_1 \| p_2) = \frac{1}{n_1} \sum_{i=1}^{n_1} \log \frac{(n_1 - 1)\mathrm{dist}_1(\boldsymbol{x}_i)^{d_{\mathrm{x}}}}{n_2 \mathrm{dist}_2(\boldsymbol{x}_i)^{d_{\mathrm{x}}}}$$

によって，$\mathrm{KL}(p_1 \| p_2)$ が推定される．上式における $\mathrm{dist}_1(\boldsymbol{x})$ と $\mathrm{dist}_2(\boldsymbol{x})$ は $\boldsymbol{x}$ からそれぞれ $\mathcal{X}_1$ と $\mathcal{X}_2$ 内の最近傍データ点への距離を示す．しかしながら，$\widehat{\mathrm{KL}}(p_1 \| p_2)$ は，次のような量に比例したバイアスをもつが知られている [4]．

$$\frac{\mathrm{tr}(\nabla\nabla p_1)}{((n_1 - 1)p_1)^{2/d_{\mathrm{x}}} p_1} - \frac{\mathrm{tr}(\nabla\nabla p_2)}{(n_2 p_2)^{2/d_{\mathrm{x}}} p_2}$$

tr は対角和，$\nabla\nabla p_1$ は，p_1 のヘッセ行列を表す．このバイアスを削減するために，[5, 8] では，再スケールしたバイアス行列

$$\tilde{\boldsymbol{B}} = \frac{1}{(n_1 - 1)^{2/d}} \left(\frac{p_2}{p_1}\right)^{2/d_{\mathrm{x}}+1} \nabla\nabla p_1 - \frac{1}{n_2^{2/d_{\mathrm{x}}}} \nabla\nabla p_2$$

を推定し，計量学習によって，$\widehat{\mathrm{KL}}(p_1 \| p_2)$ の推定バイアスを削減する方法が提案されている．$\tilde{\boldsymbol{B}}$ を推定する際には，ヘシアン行列 $\nabla\nabla p_1, \nabla\nabla p_2$ は ISED によって，密度比 p_2/p_1 は [26] で提案された密度比推定法を用いている．バイアス削減を施したカルバック・ライブラー情報量推定法は，既存の手法を上回る推定性能をもつことが示されている [5, 8]．

4.2 LSLDG の拡張について

LSLDG の拡張として，マルチタスク学習を適用した拡張がある [27]．マルチタスク学習の目的は，複数のタスクが与えられたとき，タスク間の類似性を活用することで学習精度を向上させることである [28]．LSLDG への応用では，各次元 j に対する $\partial_j \log p(\boldsymbol{x})$ の推定を 1 つのタスクとみなすと，$\partial_j \log p(\boldsymbol{x})$ は共通の $\log p(\boldsymbol{x})$ より計算されるため，タスク間に類似性が存在する．次の関係

$$\partial_j \log p(\boldsymbol{x}) \approx \widehat{r}^{(j)}(\boldsymbol{x}) = \sum_{i=1}^{b} \widehat{\theta}_{ij} \partial_j \phi_i(\boldsymbol{x}) = \partial_j \sum_{i=1}^{b} \widehat{\theta}_{ij} \phi_i(\boldsymbol{x}) = \partial_j \widehat{\boldsymbol{\theta}}_j^{\top} \boldsymbol{\phi}(\boldsymbol{x})$$

より，$\widehat{\boldsymbol{\theta}}_j^{\top} \boldsymbol{\phi}(\boldsymbol{x})$ が $\log p(\boldsymbol{x})$ の近似とみなせるため，$\widehat{\boldsymbol{\theta}}_1 \approx \widehat{\boldsymbol{\theta}}_2 \approx \cdots \approx \widehat{\boldsymbol{\theta}}_d$ となるように，$\boldsymbol{\theta}_j$ を推定することで，タスク間の類似性を活用できる．具体的には，正則化マルチタスク学習 [29, 30] を適用するために，次のような正則化項をさらに加える．

$$\gamma \sum_{j=1}^{d_x} \sum_{j'=1}^{d_x} \|\boldsymbol{\theta}_j - \boldsymbol{\theta}_{j'}\|^2 \tag{15}$$

γ は正則化パラメータである．正則化項 (15) より，$\gamma \to \infty$ のとき，$\boldsymbol{\theta}_1 = \boldsymbol{\theta}_2 = \cdots = \boldsymbol{\theta}_d$ となることが分かる．そして，正則化マルチタスク学習によって拡張された LSLDG は，数値実験より，データ点数 n が小さいときに有効であることが示された．

他の LSLDG の拡張では，データがリーマン多様体に属している場合を考える [31]．例えば，画像工学の分野で，動画内の物体の運動を取り扱う際に，物体の特徴点がある多様体に属することがしばしば仮定される [32]．LSLDG では，ユークリッド距離を用いているため，データが非ユークリッド空間内に存在する場合は，必ずしも優れた性能が得られるとは限らない．リーマン多様体上のデータ点を扱うために，[31] では，測地距離を用いて，LSLDG を拡張し，さらに，モード探索クラスリングへ応用している．グラスマン多様体に属するデータを用いた数値実験によって，そのクラスタリング法が，LSLDGC を大きく上回る性能を示すことが確認されている．

4.3 非ガウス成分分析への応用

非ガウス成分分析（non-Gaussian component analysis: NGCA） [9] は，教師無し線形次元削減の枠組みであり，射影後の入力データが非ガウス分布に従うような部分空間を見つけることを目的としている．非ガウス成分分析では，最初に，非ガウス分布に従う

入力： 入力データ $\{\boldsymbol{x}_i\}_{i=1}^n$.

ステップ 1 $\boldsymbol{x}_i$ を中心化した後で，白色化.

ステップ 2 LSLDG を用いて，白色化後のデータ $\{\boldsymbol{y}_i(=\ \widehat{\boldsymbol{\Sigma}}^{-1/2}\boldsymbol{x}_i)\}_{i=1}^n$ から，$\nabla_{\boldsymbol{y}} \log p(\boldsymbol{y})$ を推定.

ステップ 3 推定された勾配 $\widehat{\boldsymbol{r}}(\boldsymbol{y}_i)$ から，$\widehat{\boldsymbol{\Gamma}} = \frac{1}{n}\sum_{i=1}^n \{\widehat{\boldsymbol{r}}(\boldsymbol{y}_i)+\boldsymbol{y}_i\}\{\widehat{\boldsymbol{r}}(\boldsymbol{y}_i)+\boldsymbol{y}_i\}^\top$ を計算.

ステップ 4 $\widehat{\boldsymbol{\Gamma}}$ に対して固有値分解を実行．上位 d_{s} 個の固有値に対応する固有ベクトルより部分空間 $\widehat{\mathcal{I}}$ を構築.

出力： $\widehat{\mathcal{L}} = \widehat{\boldsymbol{\Sigma}}^{-1/2}\widehat{\mathcal{I}}$.

図 6 LSNGCA のアルゴリズム.

確率変数 $s^{(j)}$ を含むベクトル $\boldsymbol{s} = (s^{(1)}, s^{(2)}, \ldots, s^{(d_{\mathrm{s}})})^\top$ と d_{x} 行 d_{s} 列の行列 $\boldsymbol{A}$ を用いて， 入力データ $\boldsymbol{x}$ がモデル

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s} + \boldsymbol{n}$$

によって生成されると仮定する．$\boldsymbol{n}$ は平均 $\boldsymbol{0}$，共分散行列 $\boldsymbol{C}$ のガウスノイズである．この仮定の下で，確率密度関数 $p(\boldsymbol{x})$ が次のようなセミパラメトリックモデルで表現されることが証明されている [9].

$$p(\boldsymbol{x}) = f_{\mathrm{x}}(\boldsymbol{W}^\top \boldsymbol{x})\mathcal{N}(\boldsymbol{0}, \boldsymbol{C}) \tag{16}$$

式 (16) における $\boldsymbol{W}$ は d_{x} 行 d_{s} 列の行列，f_{x} は非負の関数である．式 (16) では，$\boldsymbol{W}, f_{\mathrm{x}}, \boldsymbol{C}$ を一意に決定できないが，次のような部分空間は同定可能である [33].

$$\mathcal{L} = \mathrm{Ker}(\boldsymbol{W}^\top)^\perp = \mathrm{Range}(\boldsymbol{W}) \tag{17}$$

ここで，Ker と Range は零空間と値域を表し，$^\perp$ は直交補空間を表す．上式における $\mathcal{L}$ は非ガウス部分空間と呼ばれる部分空間である．非ガウス成分分析における問題は，$\{\boldsymbol{x}_i\}_{i=1}^n$ から $\mathcal{L}$ を推定することである.

非ガウス成分分析においても LSLDG を活用した計算効率の良いアルゴリズムを導出することができる [10]．入力データを白色化することで，セミパラメトリックモデル (16) は次のような簡略化した形式となる [34].

$$p(\boldsymbol{y}) = f_{\mathrm{y}}(\boldsymbol{W}'^\top \boldsymbol{y})\mathcal{N}(\boldsymbol{0}, \mathbf{I}_{d_{\mathrm{x}}}) \tag{18}$$

$\boldsymbol{y} = \boldsymbol{\Sigma}^{-1/2}\boldsymbol{x}$ は白色化後のデータ，$\boldsymbol{\Sigma} = E\{\boldsymbol{x}\boldsymbol{x}^\top\}$，$f_{\mathrm{y}}$ は非負の関数，$\boldsymbol{W}'$ は $\boldsymbol{W}'^\top\boldsymbol{W}' = \mathbf{I}_{d_{\mathrm{s}}}$ を満たす d_{x} 行 d_{s} 列の行列である．元のセミパラメトリックモデル (16) との最大の違いは，$\mathcal{N}$ 内の共分散行列が単位行列 $\mathbf{I}_{d_{\mathrm{x}}}$ へと変換されたことである．簡略化されたセミパラメトリックモデル (18) の下で，推定したい部分空間は，$\mathcal{L} = \mathrm{Range}(\boldsymbol{W}) = \boldsymbol{\Sigma}^{-1/2}\mathrm{Range}(\boldsymbol{W}')$ となる．$\boldsymbol{\Sigma}$ は，$\widehat{\boldsymbol{\Sigma}} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i\boldsymbol{x}_i^\top$ によって推定可能なので，$\mathrm{Range}(\boldsymbol{W}')$ を何らかの方法で推定する必要がある．

$\mathrm{Range}(\boldsymbol{W}')$ を推定するために，簡略化されたセミパラメトリックモデル (18) から次の関係が得られる．

$$\nabla_{\boldsymbol{y}}\left[\log p(\boldsymbol{y}) - \log\mathcal{N}(\mathbf{0}, \mathbf{I}_{d_{\mathrm{x}}})\right] = \boldsymbol{W}'\nabla_{\boldsymbol{W}'^\top\boldsymbol{y}}\log f_{\mathrm{y}}(\boldsymbol{W}'^\top\boldsymbol{y}) \tag{19}$$

式 (19) は，左辺 $\nabla_{\boldsymbol{y}}\left[\log p(\boldsymbol{y}) - \log\mathcal{N}(\mathbf{0}, \mathbf{I}_{d_{\mathrm{x}}})\right] = \nabla_{\boldsymbol{y}}\log p(\boldsymbol{y}) + \boldsymbol{y}$ が，$\mathrm{Range}(\boldsymbol{W}')$ に存在することを意味する．したがって，$\boldsymbol{\eta} = \nabla_{\boldsymbol{y}}\log p(\boldsymbol{y}) + \boldsymbol{y}$ とすると，主成分分析と同様に，$\mathrm{Range}(\boldsymbol{W}')$ は，$E\{\boldsymbol{\eta}\boldsymbol{\eta}^\top\}$ の上位 d_{s} 個の固有値に対応した固有ベクトルが張る部分空間として推定することができる．

この考えに基づいた手法である**最小 2 乗非ガウス成分分析**（least-squares NGCA: LSNGCA）[10] のアルゴリズムが，図 6 に示されている．LSNGCA は，数値実験によって，既存の NGCA の手法と比較して，推定精度と計算効率性が良いことも示されている [10]．しかしながら，推定された共分散行列 $\widehat{\boldsymbol{\Sigma}}$ の条件数が大きいとき，LSNGCA における白色化ステップは大きな問題となる．この問題に対して，[35] では，元のセミパラメトリックモデル (16) から出発し，LSNGCA を白色化ステップを含まない手法へ発展させ，更に非ガウス部分空間の推定精度を向上させている．

5 まとめ

本稿では，確率密度関数の微分を直接推定するいくかの手法について解説・紹介した．これら手法は，多くの機械学習問題に適用可能であるため，今後のさらなる発展が期待できる．提案手法の理論的な解析は興味深い研究課題であり，今後の重要課題の 1 つである．

謝辞

本稿の研究において，佐々木博昭は，JSPS 科研費 15H06103 の助成を受けた．杉山将は，JST CREST の助成を受けた．

参考文献

[1] K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21(1):32–40, 1975.

[2] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799, 1995.

[3] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

[4] Y. K. Noh, M. Sugiyama, S. Liu, M. C. du Plessis, F. C. Park, and D. D. Lee. Bias reduction and metric learning for nearest-neighbor estimation of Kullback-Leibler divergence. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 669–677, 2014.

[5] H. Sasaki, Y. K. Noh, and M. Sugiyama. Direct density-derivative estimation and its application in KL-divergence approximation. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 809–818, 2015.

[6] B.W. Silverman. *Density Estimation for Statistics and Data Analysis.* CRC press, 1986.

[7] M. P. Wand and M. C. Jones. Multivariate plug-in bandwidth selection. *Computational Statistics*, 9(2):97–116, 1994.

[8] S. Sasaki, Y. K. Noh, G. Niu, and M. Sugiyama. Direct density derivative estimation. *Neural Computation.* to appear.

[9] G. Blanchard, M. Kawanabe, M. Sugiyama, V. Spokoiny, and K.R. Müller. In search of non-Gaussian components of a high-dimensional distribution. *Journal of Machine Learning Research*, 7:247–282, 2006.

[10] H. Sasaki, G. Niu, and M. Sugiyama. Non-Gaussian component analysis with log-density gradient estimation. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2016. to appear.

[11] K.C. Li. Sliced inverse regression for dimension reduction. *Journal of the American Statistical Association*, 86(414):316–327, 1991.

[12] H. Sasaki, V. Tangkaratt, and M. Sugiyama. Sufficient dimension reduction via direct estimation of the gradients of logarithmic conditional densities. In *Proceedings of the 7th Asian Conference on Machine Learning (ACML)*, volume 45, pages 33–48, 2015.

[13] R.S. Singh. Applications of estimators of a density and its derivatives to certain statistical problems. *Journal of the Royal Statistical Society. Series B*, 39(3):357–363, 1977.

[14] H. Sasaki, A. Hyvärinen, and M. Sugiyama. Clustering via mode seeking by direct estimation of the gradient of a log-density. In *Machine Learning and Knowledge Discovery in Databases Part III- European Conference, ECML/PKDD 2014*, volume 8726, pages 19–34, 2014.

[15] D. D. Cox. A penalty method for nonparametric estimation of the logarithmic derivative of a density function. *Annals of the Institute of Statistical Mathematics*, 37(1):271–288, 1985.

[16] C.M. Stein. Estimation of the mean of a multivariate normal distribution. *The annals of Statistics*, pages 1135–1151, 1981.

[17] A. Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6:695–709, 2005.

[18] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985.

[19] R. D. Cook. *Regression Graphics: Ideas for Studying Regressions Through Graphics*. John Wiley & Sons, 1998.

[20] F. Chiaromonte, R. D. Cook, and B. Li. Sufficient dimension reduction in regressions with categorical predictors. *The Annals of Statistics*, pages 475–497, 2002.

[21] K. Fukumizu, F. R. Bach, and M. I. Jordan. Dimensionality reduction for supervised learning with reproducing kernel hilbert spaces. *Journal of Machine Learning Research*, 5:73–99, 2004.

[22] T. Suzuki and M. Sugiyama. Sufficient dimension reduction via squared-loss mutual information estimation. *Neural Computation*, 25(3):725–758, 2013.

[23] Y. Xia. A constructive approach to the estimation of dimension reduction directions. *The Annals of Statistics*, 35(6):2654–2690, 2007.

[24] Jianqing Fan, Qiwei Yao, and Howell Tong. Estimation of conditional densities

and sensitivity measures in nonlinear dynamical systems. *Biometrika*, 83(1):189–206, 1996.

[25] Q. Wang, S. R. Kulkarni, and S. Verdu. A nearest-neighbor approach to estimating divergence between continuous random vectors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 55(5):2392–2405, 2006.

[26] T. Kanamori, S. Hido, and M. Sugiyama. A least-squares approach to direct importance estimation. *Journal of Machine Learning Research*, 10:1391–1445, 2009.

[27] I. Yamane, H. Sasaki, and M. Sugiyama. Regularized multi-task learning for multi-dimensional log-density gradient estimation. *Neural Computation*. to appear.

[28] R. Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

[29] T. Evgeniou and M. Pontil. Regularized multi–task learning. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 109–117. ACM, 2004.

[30] A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 41–48, Cambridge, MA, 2007. MIT Press.

[31] M. Ashizawa, H. Sasaki, Sakai T., and M. Sugiyama. Least-squares log-density gradient clustering for riemannian manifolds. *IEICE Tech. Rep.*, 115(511):17–24, 2016.

[32] V.M. Govindu. Lie-algebraic averaging for globally consistent motion estimation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2004. CVPR 2004.*, volume 1, pages 684–691. IEEE, 2004.

[33] F.J. Theis and M. Kawanabe. Uniqueness of non-Gaussian subspace analysis. In *Proceedings of the sixth international conference on Independent Component Analysis and Blind Signal Separation*, volume 3889, pages 917–925. 2006.

[34] M. Sugiyama, M. Kawanabe, G. Blanchard, and K.R. Müller. Approximating the best linear unbiased estimator of non-Gaussian signals with Gaussian noise. *IEICE transactions on information and systems*, 91(5):1577–1580, 2008.

[35] H. Shiino, H. Sasaki, G. Niu, and M. Sugiyama. Whitening-free least-squares non-Gaussian component analysis. *arXiv preprint arXiv:1603.01029*, 2016.