Non-Gaussian Component Analysis with Log-Density Gradient Estimation

Hiroaki Sasaki
Grad. School of Info. Sci.
Nara Institute of Sci. & Tech.
Nara, Japan
hsasaki@is.naist.jp

Gang Niu
Grad. School of Frontier Sci.
The University of Tokyo
Tokyo, Japan
gang@ms.k.u-tokyo.ac.jp

Masashi Sugiyama
Grad. School of Frontier Sci.
The University of Tokyo
Tokyo, Japan
sugi@k.u-tokyo.ac.jp

Abstract

Non-Gaussian component analysis (NGCA) is aimed at identifying a linear subspace such that the projected data follows a non-Gaussian distribution. In this paper, we propose a novel NGCA algorithm based on log-density gradient estimation. Unlike existing methods, the proposed NGCA algorithm identifies the linear subspace by using the eigenvalue decomposition without any iterative procedures, and thus is computationally reasonable. Furthermore, through theoretical analysis, we prove that the identified subspace converges to the true subspace at the optimal parametric rate. Finally, the practical performance of the proposed algorithm is demonstrated on both artificial and benchmark datasets.

1 Introduction

A popular way to alleviate difficulties in statistical data analysis is to reduce the dimensionality of data. Real-world applications imply that a small number of non-Gaussian signal components in data often include “interesting” information, while the remaining Gaussian components are “uninteresting” (Blanchard et al., 2006). This is the fundamental motivation of non-Gaussian-based unsupervised dimension reduction methods.

A well-known method is projection pursuit (PP), which estimates directions on which the projected data is as non-Gaussian as possible (Friedman and Tukey, 1974; Huber, 1985). In practice, PP algorithms maximize a *single* index function measuring non-Gaussianity of the data projected on a direction. However, some index functions are suitable for measuring super-Gaussianity, while others are good at measuring sub-Gaussianity (Hyvärinen et al., 2001). Thus, PP algorithms might not work well when super- and sub-Gaussian signal components are mixtured in data.

Non-Gaussian component analysis (NGCA) (Blanchard et al., 2006) copes with this problem. NGCA is a semi-parametric framework for unsupervised linear dimension reduction, and aimed at identifying a subspace such that the projected data follows a non-Gaussian distribution. Compared with independent component analysis (ICA) (Comon, 1994; Hyvärinen et al., 2001), NGCA stands on a more general setting: There is no restriction about the number of Gaussian components and non-Gaussian signal components can be dependent of each other, while ICA makes a stronger assumption that *at most* one Gaussian component is allowed and all the signal components are statistically independent of each other.

To take into account both super- and sub-Gaussian components, the first practical NGCA algorithm called the *multi-index projection pursuit* (MIPP) heuristically makes use of multiple index functions in PP (Blanchard et al., 2006), but it seems unclear whether this heuristic works well in general. To improve the performance of MIPP, *iterative metric adaptation for radial kernel functions* (IMAK) has been proposed (Kawanabe et al., 2007). IMAK does not rely on index functions, but instead estimates alternative functions from data. However, IMAK involves an iterative optimization procedure, and its computational cost is expensive.

In this paper, based on log-density gradient estimation, we propose a novel NGCA algorithm which we call the least-squares NGCA (LSNGCA). The rationale in LSNGCA is that as we show later, the target

Appearing in Proceedings of the 19^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2016, Cadiz, Spain. JMLR: W&CP volume 51. Copyright 2016 by the authors.

subspace contains the log-gradient for the data density subtracted by the log-gradient for a Gaussian density. Thus, the subspace can be identified using the eigenvalue decomposition. Unlike MIPP and IMAK, LSNGCA neither requires index functions nor any iterative procedures, and thus is computationally reasonable.

A technical challenge in LSNGCA is to accurately estimate the gradient of the log-density for data. To overcome it, we employ a direct estimator called the *least squares log-density gradients* (LSLDG) (Cox, 1985; Sasaki et al., 2014). LSLDG accurately and efficiently estimates log-density gradients in a closed form without going through density estimation. In addition, it includes an automatic parameter tuning method. In this paper, based on LSLDG, we theoretically prove that the subspace identified by LSNGCA converges to the true subspace at the optimal parametric rate, and finally demonstrate that LSNGCA reasonably works well on both artificial and benchmark datasets.

This paper is organized as follows: In Section 2, after stating the problem of NGCA, we review MIPP and IMAK, and discuss their drawbacks. We propose LSNGCA, and then overview LSLDG in Section 3. Section 4 performs theoretical analysis of LSNGCA. The performance of LSNGCA on artificial datasets is illustrated in Sections 5. Application to binary classification on benchmark datasets is given in Section 6. Section 7 concludes this paper.

2 Review of Existing Algorithms

In this section, we first describe the problem of NGCA, and then review existing NGCA algorithms.

2.1 Problem Setting

Suppose that a number of samples $\mathcal{X} = \{\boldsymbol{x}_i = (x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(d_\mathbf{x})})^\top\}_{i=1}^n$ are generated according to the following model:

$$\boldsymbol{x} = \mathbf{A}\boldsymbol{s} + \boldsymbol{n}, \tag{1}$$

where $\boldsymbol{s} = (s^{(1)}, s^{(2)}, \dots, s^{(d_\mathbf{s})})^\top$ denotes a random signal vector, $\mathbf{A}$ is a $d_\mathbf{x}$-by-$d_\mathbf{s}$ matrix, $\boldsymbol{n}$ is a Gaussian noise vector with the mean vector $\mathbf{0}$ and covariance matrix $\mathbf{C}$. Assume further that the dimensionality of $\boldsymbol{s}$ is lower than that of $\boldsymbol{x}$, namely $d_\mathbf{s} < d_\mathbf{x}$, and $\boldsymbol{s}$ and $\boldsymbol{n}$ are statistically independent of each other.

Lemma 1 in Blanchard et al. (2006) states that when data samples follow the generative model (1), the probability density $p(\boldsymbol{x})$ can be described as a semi-parametric model:

$$p(\boldsymbol{x}) = f_\mathbf{x}(\mathbf{B}^\top\boldsymbol{x})\phi_\mathbf{C}(\boldsymbol{x}), \tag{2}$$

where $\mathbf{B}$ is a $d_\mathbf{x}$-by-$d_\mathbf{s}$ matrix, $f_\mathbf{x}$ is a positive function and $\phi_\mathbf{C}$ denotes the Gaussian density with the mean $\mathbf{0}$ and covariance matrix $\mathbf{C}$.

The decomposition in (2) is not unique because $f_\mathbf{x}$, $\mathbf{B}$ and $\mathbf{C}$ are not identifiable from p. However, as shown in Theis and Kawanabe (2006), the following linear $d_\mathbf{s}$-dimensional subspace is identifiable:

$$\mathcal{L} = \mathrm{Ker}(\mathbf{B}^\top)^\perp = \mathrm{Range}(\mathbf{B}). \tag{3}$$

$\mathcal{L}$ is called the *non-Gaussian index space*. Here, the problem is to identify $\mathcal{L}$ from $\mathcal{X}$. In this paper, we assume that $d_\mathbf{s}$ is known.

2.2 Multi-Index Projection Pursuit

The first algorithm of NGCA called the *multi-index projection pursuit* (MIPP) was proposed based on the following key result (Blanchard et al., 2006):

Proposition 1. *Let $\boldsymbol{x}$ be a random variable whose density $p(\boldsymbol{x})$ has the semi-parametric form* (2)*, and suppose that $h(\boldsymbol{x})$ is a smooth real function on $\mathbb{R}^{d_\mathbf{x}}$. Denoting by $\mathbf{I}_{d_\mathbf{x}}$ the $d_\mathbf{x}$-by-$d_\mathbf{x}$ identity matrix, assume further that $\mathbb{E}\{\boldsymbol{x}\} = \mathbf{0}$ and $\mathbb{E}\{\boldsymbol{x}\boldsymbol{x}^\top\} = \mathbf{I}_{d_\mathbf{x}}$. Then, under mild regularity conditions on h, the following $\boldsymbol{\beta}(h)$ belongs to the target space $\mathcal{L}$:*

$$\boldsymbol{\beta}(h) = \mathbb{E}\{\boldsymbol{x}h(\boldsymbol{x}) - \nabla_{\boldsymbol{x}}h(\boldsymbol{x})\},$$

where $\nabla_{\boldsymbol{x}}$ is the differential operator with respect to $\boldsymbol{x}$.

The condition that $\mathbb{E}\{\boldsymbol{x}\boldsymbol{x}^\top\} = \mathbf{I}_{d_\mathbf{x}}$ seems to be strong, but in practice it can be satisfied by whitening data. Based on Proposition 1, after whitening data samples as $\boldsymbol{y}_i = \widehat{\boldsymbol{\Sigma}}^{-1/2}\boldsymbol{x}_i$ where $\widehat{\boldsymbol{\Sigma}} = \frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i\boldsymbol{x}_i^\top$, for a bunch of functions $\{h_k\}_{k=1}^K$, MIPP estimates $\boldsymbol{\beta}(h_k) = \boldsymbol{\beta}_k$ as

$$\widehat{\boldsymbol{\beta}}_k = \frac{1}{n}\sum_{i=1}^n \boldsymbol{y}_i h_k(\boldsymbol{y}_i) - \nabla_{\boldsymbol{y}} h_k(\boldsymbol{y}_i). \tag{4}$$

Then, MIPP applies PCA to $\{\widehat{\boldsymbol{\beta}}_k\}_{k=1}^K$ and estimates $\mathcal{L}$ by pulling back the $d_\mathbf{s}$-dimensional space spanned by the first $d_\mathbf{s}$ principal directions into the original (non-whitened) space.

Although the basic procedure of MIPP is simple, there are two implementation issues: normalization of $\widehat{\boldsymbol{\beta}}_k$ and choice of functions h_k. The normalization issue comes from the fact that since (4) is a linear mapping, $\widehat{\boldsymbol{\beta}}_k$ with larger norm can be dominant in the PCA step, and they are not necessarily informative in practice. To cope with this problem, Blanchard et al. (2006) proposed the following normalization scheme:

$$\frac{\widehat{\boldsymbol{\beta}}_k}{\sqrt{\sum_{i=1}^n \|\boldsymbol{y}_i h_k(\boldsymbol{y}_i) - \nabla_{\boldsymbol{y}} h_k(\boldsymbol{y}_i)\|^2 - \|\widehat{\boldsymbol{\beta}}_k\|^2}}. \tag{5}$$

After normalization, since the squared norm of each vector is proportional to its signal-to-noise ratio, longer vectors are more informative.

MIPP is supported by theoretical analysis (Blanchard et al., 2006, Theorem 3), but the practical performance strongly depends on the choice of h. To find an informative h, the form was restricted as

$$h_{r,\boldsymbol{\omega}}(\boldsymbol{y}) = r(\boldsymbol{\omega}^\top \boldsymbol{y}),$$

where $\boldsymbol{\omega} \in \mathbb{R}^{d_{\mathbf{x}}}$ denotes a unit-norm vector, and r is some function. As a heuristic, the FastICA algorithm (Hyvärinen, 1999) was employed to find a good $\boldsymbol{\omega}$. Although MIPP was numerically demonstrated to outperform PP algorithms, it is unclear whether these heuristic restriction and preprocessing work well in general.

2.3 Iterative Metric Adaptation for Radial Kernel Functions

To improve the performance of MIPP, the *iterative metric adaptation for radial kernel functions* (IMAK) estimates h by directly maximizing the informative normalization criterion, which is the squared norm of (5) used for normalization in MIPP (Kawanabe et al., 2007). To estimate h, a linear-in-parameter model is used as

$$\begin{aligned} h_{\sigma^2,\mathbf{M},\boldsymbol{\alpha}}(\boldsymbol{y}) &= \sum_{i=1}^{n} \alpha_i \exp\left\{-\frac{1}{2\sigma^2}(\boldsymbol{y}-\boldsymbol{y}_i)^\top \mathbf{M}(\boldsymbol{y}-\boldsymbol{y}_i)\right\} \\ &= \sum_{i=1}^{n} \alpha_i k_{\sigma^2,\mathbf{M}}(\boldsymbol{y},\boldsymbol{y}_i), \end{aligned}$$

where $\boldsymbol{y} = \boldsymbol{\Sigma}^{-1/2}\boldsymbol{x}$, $\boldsymbol{\Sigma} = \mathbb{E}\{\boldsymbol{x}\boldsymbol{x}^\top\}$, $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$ is a vector of parameters to be estimated, $\mathbf{M}$ is a positive semidefinite matrix and σ is a fixed scale parameter. This model allows us to represent the squared norm of the informative criterion (5) as

$$\frac{\|\widehat{\boldsymbol{\beta}}_k\|^2}{\sum_{i=1}^{n}\|\boldsymbol{y}_i h_k(\boldsymbol{y}_i) - \nabla_{\boldsymbol{y}} h_k(\boldsymbol{y}_i)\|^2 - \|\widehat{\boldsymbol{\beta}}_k\|^2} = \frac{\boldsymbol{\alpha}^\top \mathbf{F}\boldsymbol{\alpha}}{\boldsymbol{\alpha}^\top \mathbf{G}\boldsymbol{\alpha}}. \tag{6}$$

$\mathbf{F}$ and $\mathbf{G}$ in (6) are given by

$$\mathbf{F} = \frac{1}{n^2}\sum_{r=1}^{d_{\mathbf{x}}} \left(\boldsymbol{e}_r^\top \mathbf{Y}\mathbf{K} - \mathbf{1}_n^\top \partial_r \mathbf{K}\right)^\top \left(\boldsymbol{e}_r^\top \mathbf{Y}\mathbf{K} - \mathbf{1}_n^\top \partial_r \mathbf{K}\right)$$

$$\begin{aligned} &\mathbf{G} + \mathbf{F} \\ &= \frac{1}{n}\sum_{r=1}^{d_{\mathbf{x}}} \left\{\mathrm{diag}(\boldsymbol{e}_r^\top \mathbf{Y})\mathbf{K} - \partial_r \mathbf{K}\right\}^\top \left\{\mathrm{diag}(\boldsymbol{e}_r^\top \mathbf{Y})\mathbf{K} - \partial_r \mathbf{K}\right\}, \end{aligned}$$

where $\boldsymbol{e}_r$ denotes the r-th basis vector in $\mathbb{R}^{d_{\mathbf{x}}}$, $\mathbf{Y}$ is a $d_{\mathbf{x}}$-by-n matrix whose column vectors are $\boldsymbol{y}_i$, $\mathbf{K}$ is the Gram matrix whose (i,j)-th element is $[\mathbf{K}]_{ij} = k_{\sigma^2,\mathbf{M}}(\boldsymbol{y}_i, \boldsymbol{y}_j)$, ∂_r denotes the partial derivative with respect to the r-th coordinate in $\boldsymbol{y}$, and

$$\begin{aligned} [\partial_r \mathbf{K}]_{ij} = \frac{1}{\sigma^2}&\left([\mathbf{M}\boldsymbol{y}_i]_r - [\mathbf{M}\boldsymbol{y}_j]_r\right) \\ &\times k'_{\sigma^2,\mathbf{M}}\left(-\frac{1}{2\sigma^2}(\boldsymbol{y}_i - \boldsymbol{y}_j)^\top \mathbf{M}(\boldsymbol{y}_i - \boldsymbol{y}_j)\right). \end{aligned}$$

The maximizer of (6) can be obtained by solving the following generalized eigenvalue problem:

$$\mathbf{F}\boldsymbol{\alpha} = \eta \mathbf{G}\boldsymbol{\alpha},$$

where η is the generalized eigenvalue. Once $\boldsymbol{\alpha}$ is estimated, $\boldsymbol{\beta}$ can be also estimated according to (4). Then, the metric $\mathbf{M}$ in h is updated as

$$\mathbf{M} \propto \sum_k \widehat{\boldsymbol{\beta}}_k \widehat{\boldsymbol{\beta}}_k^\top,$$

where $\mathbf{M}$ is scaled so that its trace equals to $d_{\mathbf{x}}$. IMAK alternately and iteratively updates $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. It was experimentally shown that IMAK improves the performance of MIPP. However, IMAK makes use of the above alternate and iterative procedure to estimate a number of functions $h_{\sigma^2,\mathbf{M},\boldsymbol{\alpha}}$ with different parameter values for σ. Thus, IMAK is computationally costly.

3 Least-Squares Non-Gaussian Component Analysis (LSNGCA)

In this section, we propose a novel algorithm for NGCA, which is based on the gradients of log-densities. Then, we overview an existing useful estimator for log-density gradients.

3.1 A Log-Density-Gradient-Based Algorithm for NGCA

In contrast to MIPP and IMAK, our algorithm does not rely on Proposition 1, but is derived more directly from the semi-parametric model (2). As stated before, the noise covariance matrix $\mathbf{C}$ in (2) cannot be identified in general. However, after whitening data, the semi-parametric model (2) is significantly simplified by following the proof of Lemma 3 in Sugiyama et al. (2008) as

$$p(\boldsymbol{y}) = f_{\mathbf{y}}(\mathbf{B}'^\top \boldsymbol{y})\phi_{\mathbf{I}_{d_{\mathbf{x}}}}(\boldsymbol{y}), \tag{7}$$

where $f_{\mathbf{y}}$ is a positive function, and $\mathbf{B}'$ is a $d_{\mathbf{x}}$-by-$d_{\mathbf{s}}$ matrix such that $\mathbf{B}'^\top \mathbf{B}' = \mathbf{I}_{d_{\mathbf{s}}}$. Thus, under (7), the non-Gaussian index subspace can be represented as $\mathcal{L} = \mathrm{Range}(\mathbf{B}) = \boldsymbol{\Sigma}^{-1/2}\mathrm{Range}(\mathbf{B}')$.

To estimate $\mathrm{Range}(\mathbf{B}')$, we take a novel approach based on the gradients of log-densities. The reason of using the gradients comes from the following equation, which can be easily derived by computing the gradient of the both-hand sides of (7) after taking the logarithm:

$$\nabla_{\boldsymbol{y}}[\log p(\boldsymbol{y}) - \log \phi_{\mathbf{I}_{d_{\mathbf{x}}}}(\boldsymbol{y})] = \mathbf{B}'\nabla_{\boldsymbol{z}} \log f_{\mathbf{y}}(\boldsymbol{z} = \mathbf{B}'^{\top}\boldsymbol{y}). \tag{8}$$

Eq.(8) indicates that $\nabla_{\boldsymbol{y}}[\log p(\boldsymbol{y}) - \log \phi_{\mathbf{I}_{d_{\mathbf{x}}}}(\boldsymbol{y})] = \nabla_{\boldsymbol{y}} \log p(\boldsymbol{y}) + \boldsymbol{y}$ is contained in Range$(\mathbf{B}')$. Thus, an orthonormal basis $\{\boldsymbol{e}'_i\}_{i=1}^{d_{\mathbf{s}}}$ in Range$(\mathbf{B}')$ is estimated as the minimizer of the following PCA-like problem:

$$\begin{aligned}&\mathbb{E}\{\|\boldsymbol{\nu} - \sum_{i=1}^{d_{\mathbf{s}}}(\boldsymbol{\nu}^{\top}\boldsymbol{e}'_i)\boldsymbol{e}'_i\|^2\}\\ &= \mathbb{E}\{\|\boldsymbol{\nu}\|^2\} - \sum_{i=1}^{d_{\mathbf{s}}}\boldsymbol{e}_i'^{\top}\mathbb{E}\{\boldsymbol{\nu}\boldsymbol{\nu}^{\top}\}\boldsymbol{e}'_i,\end{aligned} \tag{9}$$

where $\boldsymbol{\nu} = \nabla_{\boldsymbol{y}} \log p(\boldsymbol{y}) + \boldsymbol{y}$, $\|\boldsymbol{e}_i\| = 1$ and $\boldsymbol{e}_i^{\top}\boldsymbol{e}_j = 0$ for $i \neq j$. Eq.(9) indicates that minimizing the left-hand side with respect to $\boldsymbol{e}_i$ is equivalent to maximizing the second term in the right-hand side. Thus, an orthonormal basis $\{\boldsymbol{e}_i\}_{i=1}^{d_{\mathbf{s}}}$ can be estimated by applying the eigenvalue decomposition to $\mathbb{E}\{\boldsymbol{\nu}\boldsymbol{\nu}^{\top}\} = \mathbb{E}\{(\nabla_{\boldsymbol{y}} \log p(\boldsymbol{y}) + \boldsymbol{y})(\nabla_{\boldsymbol{y}} \log p(\boldsymbol{y}) + \boldsymbol{y})^{\top}\}$.

The proposed LSNGCA algorithm is summarized in Fig.1. Compared with MIPP and IMAK, LSNGCA estimates $\mathcal{L}$ without specifying or estimating h in Proposition 1 and any iteration procedures. The key challenge in LSNGCA is to estimate log-density gradients $\nabla_{\boldsymbol{y}} \log p(\boldsymbol{y})$ in Step 2. To overcome this challenge, we employ a method called the *least-squares log-density gradients* (LSLDG) (Cox, 1985; Sasaki et al., 2014), which directly estimates log-density gradients without going through density estimation. As overviewed below, with LSLDG, LSNGCA can compute all the solutions in a closed form, and thus would be computationally efficient.

```
Input: Data samples, $\{\boldsymbol{x}_i\}_{i=1}^{n}$.
Step 1  Whiten $\boldsymbol{x}_i$ after subtracting the empirical mean values from them.
Step 2  Estimate the gradient of the log-density for the whitened data $\boldsymbol{y}_i = \widehat{\boldsymbol{\Sigma}}^{-1/2}\boldsymbol{x}_i$.
Step 3  Using the estimated gradients $\widehat{\boldsymbol{g}}(\boldsymbol{y}_i)$, compute $\widehat{\boldsymbol{\Gamma}} = \frac{1}{n}\sum_{i=1}^{n}\{\widehat{\boldsymbol{g}}(\boldsymbol{y}_i) + \boldsymbol{y}_i\}\{\widehat{\boldsymbol{g}}(\boldsymbol{y}_i) + \boldsymbol{y}_i\}^{\top}$.
Step 4  Perform the eigenvalue decomposition to $\widehat{\boldsymbol{\Gamma}}$, and let $\widehat{\mathcal{I}}$ be the space spanned by the $d_{\mathbf{s}}$ directions corresponding to the largest $d_{\mathbf{s}}$ eigenvalues.
Output: $\widehat{\mathcal{L}} = \widehat{\boldsymbol{\Sigma}}^{-1/2}\widehat{\mathcal{I}}$.
```

Figure 1: The LSNGCA algorithm.

3.2 Least-Squares Log-Density Gradients (LSLDG)

The fundamental idea of LSLDG is to directly fit a gradient model $g^{(j)}(\boldsymbol{x})$ to the true log-density gradient under the squared-loss:

$$\begin{aligned}&J(g^{(j)})\\ &= \int \left\{g^{(j)}(\boldsymbol{x}) - \partial_j \log p(\boldsymbol{x})\right\}^2 p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} - C^{(j)}\\ &= \int \left\{g^{(j)}(\boldsymbol{x})\right\}^2 p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} - 2\int g^{(j)}(\boldsymbol{x})\partial_j p(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\\ &= \int \left\{g^{(j)}(\boldsymbol{x})\right\}^2 p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} + 2\int \left\{\partial_j g^{(j)}(\boldsymbol{x})\right\} p(\boldsymbol{x})\mathrm{d}\boldsymbol{x},\end{aligned}$$

$C^{(j)} = \int \{\partial_j \log p(\boldsymbol{x})\}^2 p(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$, $\partial_j = \frac{\partial}{\partial x^{(j)}}$ and the last equality comes from the *integration by parts* under a mild assumption that $\lim_{|x^{(j)}|\to\infty} g^{(j)}(\boldsymbol{x})p(\boldsymbol{x}) = 0$. Thus, $J(g^{(j)})$ is empirically approximated as

$$\tilde{J}(g^{(j)}) = \frac{1}{n}\sum_{i=1}^{n} g^{(j)}(\boldsymbol{x}_i)^2 + 2\partial_j g^{(j)}(\boldsymbol{x}_i). \tag{10}$$

To estimate log-density gradients, we use a linear-in-parameter model as

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

where θ_{ij} is a parameter to be estimated, $\psi_{ij}(\boldsymbol{x})$ is a fixed basis function, and b denotes the number of basis functions and is fixed to $b = \min(n, 100)$ in this paper. As in Sasaki et al. (2014), the derivatives of Gaussian functions centered at $\boldsymbol{c}_i$ are used for $\psi_{ij}(\boldsymbol{x})$:

$$\psi_{ij}(\boldsymbol{x}) = \frac{[\boldsymbol{c}_i - \boldsymbol{x}]^{(j)}}{\sigma_j^2}\exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{c}_i\|^2}{2\sigma_j^2}\right),$$

where $[\boldsymbol{x}]^{(j)}$ denotes the j-th element in $\boldsymbol{x}$, σ_j is the width parameter, and the center point $\boldsymbol{c}_i$ is randomly selected from data samples $\boldsymbol{x}_i$. After substituting the linear-in-parameter model and adding the ℓ_2 regularizer into (10), the solution is computed analytically:

$$\begin{aligned}\widehat{\boldsymbol{\theta}}_j &= \underset{\boldsymbol{\theta}_j}{\operatorname{argmin}}\left[\boldsymbol{\theta}_j^{\top}\widehat{\mathbf{G}}_j\boldsymbol{\theta}_j + 2\boldsymbol{\theta}_j^{\top}\widehat{\boldsymbol{h}}_j + \lambda_j\boldsymbol{\theta}_j^{\top}\boldsymbol{\theta}_j\right]\\ &= -(\widehat{\mathbf{G}}_j + \lambda_j\mathbf{I}_b)^{-1}\widehat{\boldsymbol{h}}_j,\end{aligned}$$

where λ_j denotes the regularization parameter,

$$\widehat{\mathbf{G}}_j = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\psi}_j(\boldsymbol{x}_i)\boldsymbol{\psi}_j(\boldsymbol{x}_i)^{\top} \text{ and } \widehat{\boldsymbol{h}}_j = \frac{1}{n}\sum_{i=1}^{n}\partial_j\boldsymbol{\psi}_j(\boldsymbol{x}_i).$$

Finally, the estimator is obtained as

$$\widehat{g}^{(j)}(\boldsymbol{x}) = \widehat{\boldsymbol{\theta}}_j^\top \boldsymbol{\psi}_j(\boldsymbol{x}).$$

As overviewed, LSLDG does not perform density estimation, but directly estimates log-density gradients. The advantages of LSLDG can be summarized as follows:

- The solutions are efficiently computed in a closed form.
- All the parameters, σ_j and λ_j, can be automatically determined by cross-validation.
- Experimental results confirmed that LSLDG provides much more accurate estimates for log-density gradients than an estimator based on kernel density estimation especially for higher-dimensional data (Sasaki et al., 2014).

4 Theoretical Analysis

We investigate the convergence rate of LSNGCA in a parametric setting. Recall that

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

and denote their expectations by

$$\mathbf{G}_j^* = \mathbb{E}\left[\boldsymbol{\psi}_j(\boldsymbol{x})\boldsymbol{\psi}_j(\boldsymbol{x})^\top\right], \quad \boldsymbol{h}_j^* = \mathbb{E}\left[\partial_j \boldsymbol{\psi}_j(\boldsymbol{x})\right].$$

Subsequently, let

$$\begin{aligned}\boldsymbol{\theta}_j^* &= \mathrm{argmin}_{\boldsymbol{\theta}}\left\{\boldsymbol{\theta}^\top \mathbf{G}_j^* \boldsymbol{\theta} + 2\boldsymbol{\theta}^\top \boldsymbol{h}_j^* + \lambda_j^* \boldsymbol{\theta}^\top\boldsymbol{\theta}\right\},\\ g^{*(j)}(\boldsymbol{x}) &= \boldsymbol{\theta}_j^{*\top}\boldsymbol{\psi}_j(\boldsymbol{x}),\\ \boldsymbol{\Gamma}^* &= \mathbb{E}\left[(\boldsymbol{g}^*(\boldsymbol{y})+\boldsymbol{y})(\boldsymbol{g}^*(\boldsymbol{y})+\boldsymbol{y})^\top\right],\end{aligned}$$

let $\mathcal{I}^*$ be the eigen-space of $\boldsymbol{\Gamma}^*$ with its largest $d_{\mathbf{s}}$ eigenvalues, and $\mathcal{L}^* = \boldsymbol{\Sigma}^{-1/2}\mathcal{I}^*$ be the optimal estimate. If $\mathbf{G}_j^*$ is positive definite, $\lambda_j^* = 0$ is also allowed in our analysis by assuming the smallest eigenvalue of $\mathbf{G}_j^*$ is no less than ϵ_λ in the first condition in Theorem 1.

Theorem 1. *Given the estimated space $\widehat{\mathcal{L}}$ based on a set of data samples of size n and the optimal space $\mathcal{L}^*$, denote by $\widehat{\mathbf{E}} \in \mathbb{R}^{d_{\mathbf{x}}\times d_{\mathbf{s}}}$ the matrix form of an arbitrary orthonormal basis of $\widehat{\mathcal{L}}$ and by $\mathbf{E}^* \in \mathbb{R}^{d_{\mathbf{x}}\times d_{\mathbf{s}}}$ that of $\mathcal{L}^*$. Define the distance between spaces $\widehat{\mathcal{L}}$ and $\mathcal{L}^*$ as*

$$\mathcal{D}(\widehat{\mathcal{L}}, \mathcal{L}^*) = \inf_{\widehat{\mathbf{E}},\mathbf{E}^*} \|\widehat{\mathbf{E}} - \mathbf{E}^*\|_{\mathrm{Fro}},$$

where $\|\cdot\|_{\mathrm{Fro}}$ stands for the Frobenius norm. Then, as $n \to \infty$,

$$\mathcal{D}(\widehat{\mathcal{L}}, \mathcal{L}^*) = \mathcal{O}_p\left(n^{-1/2}\right),$$

provided that

1. *λ_j for all j converge in $\mathcal{O}(n^{-1/2})$ to the non-zero limits, i.e., $\lim_{n\to\infty} n^{1/2}|\lambda_j - \lambda_j^*| < \infty$, and there exists $\epsilon_\lambda > 0$ such that $\lambda_j^* \geq \epsilon_\lambda$;*
2. *$\psi_{ij}(\boldsymbol{x})$ for all i and j have well-chosen centers and widths, such that the first $d_{\mathbf{s}}$ eigenvalues of $\boldsymbol{\Gamma}^*$ are neither 0 nor $+\infty$.*

Theorem 1 shows that LSNGCA is consistent, and its convergence rate is $\mathcal{O}_p(n^{-1/2})$ under mild conditions. The first is about the limits of ℓ_2-regularizations, and it is easy to control. The second is also reasonable and easy to satisfy, as long as the centers are not located in regions with extremely low densities and the bandwidths are neither too large ($\widehat{\boldsymbol{\Gamma}}$ might be all-zero) nor too small ($\widehat{\boldsymbol{\Gamma}}$ might be unbounded).

Our theorem is based on two powerful theories, one is of perturbed optimizations (Bonnans and Cominetti, 1996; Bonnans and Shapiro, 1998), and the other is of matrix approximation of integral operators (Koltchinskii, 1998; Koltchinskii and Giné, 2000) that covers a theory of perturbed eigen-decompositions. According to the former, we can prove that $\widehat{\boldsymbol{\theta}}_j$ converges to $\boldsymbol{\theta}_j^*$ in $\mathcal{O}_p(n^{-1/2})$ and thus $\widehat{\boldsymbol{\Gamma}}$ to $\boldsymbol{\Gamma}^*$ in $\mathcal{O}_p(n^{-1/2})$. According to the latter, we can prove that $\widehat{\mathcal{I}}$ converges to $\mathcal{I}^*$ and therefore $\widehat{\mathcal{L}}$ to $\mathcal{L}^*$ in $\mathcal{O}_p\left(n^{-1/2}\right)$. The full proof can be found in the supplementary material.

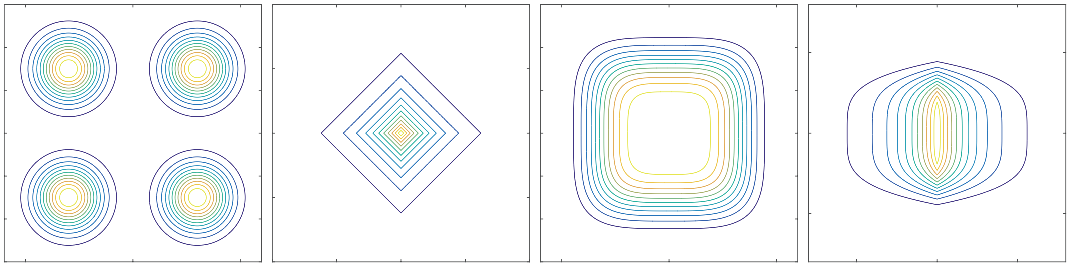
5 Illustration on Artificial Data

In this section, we experimentally illustrate how LSNGCA works on artificial data, and compare its performance with MIPP and IMAK.

Non-Gaussian signal components $\boldsymbol{s} = (s_1, s_2)^\top$ were sampled from the following distributions:

- Gaussian mixture: $p(s_1, s_2) \propto \prod_{i=1}^2 \exp\{-(s_i - 3)^2/2\} + \exp\{-(s_i+3)^2/2\}$ (Fig. 2(a)).
- Super-Gaussian: $p(s_1, s_2) \propto \prod_{i=1}^2 \exp\left(-|s_i|/\alpha\right)$ where α is determined such that the variances of s_1 and s_2 are 3 (Fig. 2(b)).
- Sub-Gaussian: $p(s_1, s_2) \propto \prod_{i=1}^2 \exp(-s_i^4/\beta)$ where β is determined such that the variances of s_1 and s_2 are 3 (Fig. 2(c)).
- Super- and sub-Gaussian: $p(s_1, s_2) = p(s_1)p(s_2)$ where $p(s_1) \propto \exp(-|s_1|/\alpha)$ and $p(s_2) \propto \exp(-s_2^4/\beta)$. α and β are determined such that the variances of s_1 and s_2 are 3 (Fig. 2(d)).

Then, a data vector was generated according to $\boldsymbol{x} = (s_1, s_2, n_3, \ldots, n_{10})$ where n_i for $i = 3, \ldots, 10$ were



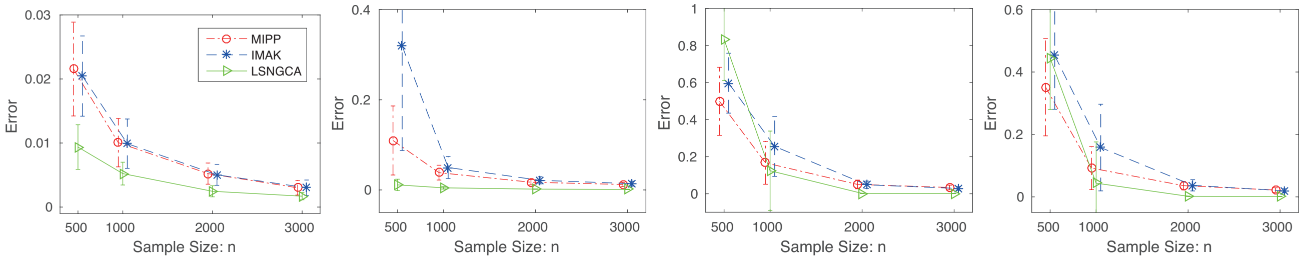
(a) Gaussian mixture

(b) Super-Gaussian

(c) Sub-Gaussian

(d) Super- and sub-Gaussian

Figure 2: The two-dimensional distributions of four non-Gaussian densities.



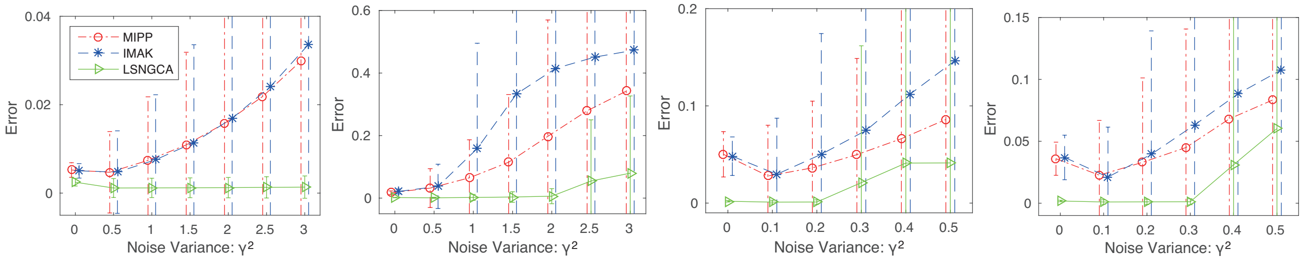
(a) Gaussian mixture

(b) Super-Gaussian

(c) Sub-Gaussian

(d) Super- and sub-Gaussian

Figure 3: The average errors over 50 runs for four kinds of non-Gaussian signal components as the functions of samples size n. The error bars denote standard deviations. The horizontal position of the markers for MIPP and IMAK was slightly modified to improve visibility of their error bars.



(a) Gaussian mixture

(b) Super-Gaussian

(c) Sub-Gaussian

(d) Super- and sub-Gaussian

Figure 4: The average errors over 50 runs for four kinds of non-Gaussian signal components as the functions of noise variances γ^2 when $n = 2,000$. The horizontal position of the markers for MIPP and IMAK was slightly modified to improve visibility of their error bars.

sampled from the independent standard normal density. The error was measured by

$$\mathcal{E}(\widehat{\mathcal{L}}, \mathcal{L}) = \frac{1}{d_{\mathbf{s}}} \sum_{i=1}^{d_{\mathbf{s}}} \|\widehat{\boldsymbol{e}}_i - \Pi_{\mathcal{L}} \widehat{\boldsymbol{e}}_i\|^2,$$

where $\{\widehat{\boldsymbol{e}}_i\}_{i=1}^{d_{\mathbf{s}}}$ is an orthonormal basis of $\widehat{\mathcal{L}}$, and $\Pi_{\mathcal{L}}$ denotes the orthogonal projection on $\mathcal{L}$. For model selection in LSLDG, a five-hold cross-validation was performed with respect to the hold-out error of (10) using the ten candidate values for σ_j (or λ_j) from 10^{-1} (or 10^{-5}) to 10 at the regular interval in logarithmic scale .

The results are presented in Fig. 3. For the Gaussian mixture and super-Gaussian cases, LSNGCA always works better than MIPP and IMAK even when the sample size is relatively small (Fig. 3(a) and (b)). On the other hand, when the signal components include sub-Gaussian components and the number of samples is insufficient, the performance of LSNGCA is not good (Fig. 3(c) and (d)). This presumably comes

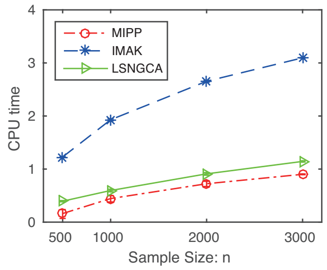


Figure 5: The average CPU time over 50 runs for the Gaussian mixture as the functions of samples size n. The vertical axis is in logarithmic scale.

from the fact that estimating the gradients for logarithmic sub-Gaussian densities is more challenging than super-Gaussian densities. However, as long as the number of samples is sufficient, the performance of LSNGCA is comparable to or slightly better than other methods.

Next, we investigate the performance of the three algorithms when the non-Gaussian signal components in data are contaminated by Gaussian noises such that $\boldsymbol{x} = (s_1 + n_1, s_2 + n_2, n_3, \ldots, n_{10})$ where n_1 and n_2 are independently sampled from the Gaussian density with the mean 0 and variance γ^2, while other n_i for $i = 3, \ldots, 10$ are sampled as in the last experiment. Fig. 4(a) and (b) show that as γ^2 increases, the estimation errors of LSNGCA for the Gaussian mixture or super-Gaussian distribution more mildly increases than MIPP and IMAK. When the data includes sub-Gaussian components, LSNGCA still works better than MIPP and IMAK for weak noise, but all methods are not robust to stronger noises.

For computational costs, MIPP is the best method, while IMAK consumes much time (Fig.5). MIPP estimates a bunch of $\boldsymbol{\beta}_k$ by simply computing (4), and FastICA used in MIPP is an iterative method, but its convergence is fast. Therefore, MIPP is a quite efficient method. As reviewed in Section 2.3, because of the alternate and iterative procedure, IMAK is computationally demanding. LSNGCA is less efficient than MIPP, but its computational time is still reasonable.

In short, LSNGCA is advantageous in terms of the sample size and noise tolerance especially when the non-Gaussian signal components follow multi-modal or super-Gaussian distributions. Furthermore, LSNGCA is not the most efficient algorithm, but its computational cost is reasonable.

6 Application to Binary Classification on Benchmark Datasets

In this section, we apply LSNGCA to binary classification on benchmark datasets. For comparison, in addition to MIPP and IMAK, we employed PCA and locality preserving projections (LPP) (He and Niyogi, 2004)[1]. For LPP, the nearest-neighbor-based weight matrix was constructed using the heat kernel whose width parameter was fixed to $t_i t_j$: t_i is the Euclidean distance to the k-nearest neighbor sample of $\boldsymbol{x}_i$. Here we set $k = 7$ as suggested by Zelnik-Manor and Perona (2005).

We used datasets for binary classification[2] which are available at `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`. For each dataset, we randomly selected n samples for the training phase, and the remaining samples were used for the test phase. For some large datasets, we randomly chose $1,000$ samples for the training phase as well as for the test phase. As preprocessing, we separately subtracted the empirical means from the training and test samples. The projection matrix was estimated from the n training samples by each method. Then, the support vector machine (SVM) (Scholkopf and Smola, 2001) was trained using the dimension-reduced training data.[3]

The averages and standard deviations for miss classification rates over 30 runs are summarized in Table 1. This table shows that LSNGCA overall compares favorably with other algorithms.

7 Conclusion

In this paper, we proposed a novel algorithm for non-Gaussian component analysis (NGCA) called the *least-squares NGCA* (LSNGCA). The subspace identification in LSNGCA is performed using the eigenvalue decomposition without any iterative procedures, and thus LSNGCA is computationally reasonable. Through theoretical analysis, we established the optimal convergence rate in a parametric setting for the subspace identification. The experimental results confirmed that LSNGCA performs better than existing algorithms especially for multi-modal or super-Gaussian signal components, and reasonably works on benchmark datasets.

[1] `http://www.cad.zju.edu.cn/home/dengcai/Data/DimensionReduction.html`

[2] The "shuttle" and "vehicle" datasets originally include samples from more than two classes. Here, we only used samples in classes 1 and 4 in the "shuttle" dataset, while we regarded samples in classes 1 and 3 as positive and others as negative in the "vehicle" dataset.

[3] We employed a MATLAB software for SVM called *LIBSVM* (Chang and Lin, 2011).

Table 1: The averages and standard deviations of misclassification rates for benchmark datasets over 30 runs. The numbers in the parentheses are standard deviations. The best and comparable methods judged by the unpaired t-test at the significance level 1% are described in boldface. The symbol "-" in the table means that IMAK unexpectedly stopped during the experiments because of a numerical problem.

	LSNGCA	MIPP	IMAK	PCA	LPP
australian $(d_{\mathbf{x}}, n) = (14, 200)$					
$d_{\mathbf{s}} = 2$	20.20(5.09)	21.02(6.66)	33.43(4.99)	**17.37(1.30)**	**17.50(1.08)**
$d_{\mathbf{s}} = 4$	**16.23(2.60)**	**15.90(2.14)**	32.53(6.06)	**14.92(1.17)**	**15.07(1.16)**
$d_{\mathbf{s}} = 6$	**15.41(2.32)**	**15.22(2.02)**	30.71(5.71)	**14.16(1.16)**	**14.39(1.10)**
breast-cancer (Bache and Lichman) $(d_{\mathbf{x}}, n) = (10, 400)$					
$d_{\mathbf{s}} = 2$	13.19(2.23)	3.73(0.92)	11.12(3.86)	**2.71(0.80)**	**2.82(0.77)**
$d_{\mathbf{s}} = 4$	9.26(1.76)	4.49(0.96)	6.50(1.55)	**2.84(0.76)**	**2.97(0.76)**
$d_{\mathbf{s}} = 6$	7.37(1.42)	4.91(0.92)	6.14(1.62)	**2.80(0.80)**	**2.97(0.73)**
cod-rna (Uzilov et al., 2006) $(d_{\mathbf{x}}, n) = (8, 200)$					
$d_{\mathbf{s}} = 2$	18.20(5.48)	32.99(2.25)	35.77(2.81)	31.44(1.88)	**14.26(1.84)**
$d_{\mathbf{s}} = 4$	**15.85(5.15)**	20.52(9.94)	33.03(1.48)	29.94(2.09)	**14.11(1.88)**
$d_{\mathbf{s}} = 6$	**13.34(4.74)**	**10.62(4.63)**	32.99(1.48)	28.81(2.45)	14.03(1.89)
diabetes (Bache and Lichman) $(d_{\mathbf{x}}, n) = (8, 400)$					
$d_{\mathbf{s}} = 2$	32.26(2.33)	33.81(1.97)	35.20(1.92)	**29.27(1.66)**	30.76(1.95)
$d_{\mathbf{s}} = 4$	32.57(2.18)	31.91(1.99)	35.64(2.02)	**26.56(1.67)**	**27.10(1.71)**
$d_{\mathbf{s}} = 6$	30.76(2.89)	29.63(1.79)	34.55(1.82)	**25.38(1.82)**	**25.82(1.87)**
liver-disorders $(d_{\mathbf{x}}, n) = (6, 200)$					
$d_{\mathbf{s}} = 2$	39.31(3.62)	**32.62(3.72)**	**33.15(5.21)**	42.14(2.71)	42.00(2.96)
$d_{\mathbf{s}} = 4$	**32.83(5.15)**	**32.02(3.67)**	35.36(3.32)	42.02(2.64)	42.02(2.71)
german.numer $(d_{\mathbf{x}}, n) = (24, 200)$					
$d_{\mathbf{s}} = 2$	**30.27(0.74)**	**30.35(0.77)**	-	**30.63(1.38)**	**30.82(1.52)**
$d_{\mathbf{s}} = 4$	**30.29(0.62)**	**30.45(0.86)**	31.12(1.22)	**29.90(1.68)**	**30.07(1.52)**
$d_{\mathbf{s}} = 6$	30.54(1.01)	30.95(0.90)	31.23(1.12)	**29.08(1.43)**	**29.46(1.09)**
SUSY $(d_{\mathbf{x}}, n) = (18, 1000)$					
$d_{\mathbf{s}} = 2$	**29.58(1.86)**	**29.42(1.70)**	34.37(1.82)	**28.71(3.11)**	35.26(1.87)
$d_{\mathbf{s}} = 4$	**25.46(2.07)**	**25.91(1.70)**	32.89(2.03)	27.05(1.55)	27.10(2.06)
$d_{\mathbf{s}} = 6$	**23.32(1.73)**	24.75(1.61)	31.74(2.16)	25.49(1.50)	25.56(1.56)
shuttle $(d_{\mathbf{x}}, n) = (9, 1000)$					
$d_{\mathbf{s}} = 2$	**11.29(2.53)**	14.39(3.34)	-	16.01(2.20)	**11.41(3.53)**
$d_{\mathbf{s}} = 4$	**6.04(3.24)**	10.45(1.12)	16.84(1.43)	8.18(0.93)	9.36(2.21)
$d_{\mathbf{s}} = 6$	**3.03(1.73)**	10.24(1.19)	16.84(1.43)	8.46(1.02)	11.03(2.91)
vehicle $(d_{\mathbf{x}}, n) = (18, 200)$					
$d_{\mathbf{s}} = 2$	**41.23(4.26)**	43.36(3.78)	49.11(2.63)	**38.88(2.47)**	46.97(2.44)
$d_{\mathbf{s}} = 4$	**35.16(3.76)**	**34.26(4.13)**	50.04(1.42)	38.43(2.16)	45.85(3.11)
$d_{\mathbf{s}} = 6$	30.72(3.95)	**26.60(2.24)**	50.33(1.19)	34.30(2.99)	45.47(4.05)
svmguide3 $(d_{\mathbf{x}}, n) = (21, 200)$					
$d_{\mathbf{s}} = 2$	**22.58(1.55)**	**23.30(1.38)**	-	**23.22(1.12)**	23.92(0.52)
$d_{\mathbf{s}} = 4$	**22.32(1.59)**	**21.63(1.28)**	23.93(0.52)	**21.74(0.92)**	23.45(0.75)
$d_{\mathbf{s}} = 6$	22.20(1.54)	**21.29(0.96)**	23.92(0.52)	22.06(0.96)	23.53(0.68)

Acknowledgements

HS did most of this work when he was working at the University of Tokyo, and was partially supported by KAKENHI 15H06103. GN thanks JST CREST. MS was partially supported by KAKENHI 25700022.

References

K. Bache and M. Lichman. UCI machine learning repository.

G. Blanchard, M. Kawanabe, M. Sugiyama,

V. Spokoiny, and K. Müller. In search of non-Gaussian components of a high-dimensional distribution. *Journal of Machine Learning Research*, 7: 247–282, 2006.

F. Bonnans and R. Cominetti. Perturbed optimization in Banach spaces I: A general theory based on a weak directional constraint qualification; II: A theory based on a strong directional qualification condition; III: Semiinfinite optimization. *SIAM Journal on Control and Optimization*, 34:1151–1171, 1172–1189, and 1555–1567, 1996.

F. Bonnans and A. Shapiro. Optimization problems with perturbations, a guided tour. *SIAM Review*, 40(2):228–264, 1998.

C. Chang and C. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3):287–314, 1994.

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.

J. Friedman and J. Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computers*, 23(9):881–890, 1974.

X. He and P. Niyogi. Locality preserving projections. In *Advances in Neural Information Processing Systems*, pages 153–160, 2004.

P. Huber. Projection pursuit. *The Annals of Statistics*, 13(2):435–475, 1985.

A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999.

A. Hyvärinen, J. Karhunen, and E. Oja. *Independent component analysis*. John Wiley & Sons, 2001.

M. Kawanabe, M. Sugiyama, G. Blanchard, and K. Müller. A new algorithm of non-Gaussian component analysis with radial kernel functions. *Annals of the Institute of Statistical Mathematics*, 59(1):57–75, 2007.

V. Koltchinskii. Asymptotics of spectral projections of some random matrices approximating integral operators. *Progress in Probabilty*, 43:191–227, 1998.

V. Koltchinskii and E. Giné. Random matrix approximation of spectra of integral operators. *Bernoulli*, 6:113–167, 2000.

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.

B. Scholkopf and A. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. The MIT press, 2001.

M. Sugiyama, M. Kawanabe, G. Blanchard, and K. 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.

F. Theis and M. Kawanabe. Uniqueness of non-Gaussian subspace analysis. In *Lecture Notes in Computer Science*, volume 3889, pages 917–925. Springer-Verlag, 2006.

A. Uzilov, J. Keegan, and D. Mathews. Detection of non-coding rnas on the basis of predicted secondary structure formation free energy change. *BMC bioinformatics*, 7(1):173, 2006.

L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In *Advances in Neural Information Processing Systems*, pages 1601–1608, 2005.