Effective Nonlinear Feature Selection Method based on HSIC Lasso and with Variational Inference

Kazuki Koyama
NTT Communications
Tokyo, Japan
kazuki.koyama@ntt.com

Keisuke Kiritoshi
NTT Communications
Tokyo, Japan
k.kiritoshi@ntt.com

Tomomi Okawachi
NTT Communications
Tokyo, Japan
t.okawachi@ntt.com

Tomonori Izumitani
NTT Communications
Tokyo, Japan
tomonori.izumitani@ntt.com

Abstract

HSIC Lasso is one of the most effective sparse nonlinear feature selection methods based on the Hilbert-Schmidt independence criterion. We propose an adaptive nonlinear feature selection method, which is based on the HSIC Lasso, that uses a stochastic model with a family of super-Gaussian prior distributions for sparsity enhancement. The method includes easily implementable closed-form update equations that are derived approximately from variational inference and can handle high-dimensional and large datasets. We applied the method to several synthetic datasets and real-world datasets and verified its effectiveness regarding redundancy, computational complexity, and classification and prediction accuracy using the selected features. The results indicate that the method can more effectively remove irrelevant features, leaving only relevant features. In certain problem settings, the method assigned non-zero importance only to the actually relevant features. This is an important characteristic for practical use.

1 INTRODUCTION

The effective selection of important features inherent in high-dimensional and large datasets is a longstanding challenge in machine learning and statistics. Generally, feature extraction creates new features from functions of the original features, whereas feature selection returns a subset of the features. Thus, many real-world applications, such as for microarray data and text mining, have also been proposed (Forman, 2008; Abusamra, 2013), since feature selection maintains the physical meanings of the original features and gives models better interpretability by keeping some of the original features.

The simplest feature selection method is to incorporate L_1 regularization into the model, such as the Least Absolute Shrinkage and Selection Operator (Lasso) (Tibshirani, 1996). Lasso-based feature selection methods are useful because they are computationally inexpensive, and feature selection is carried out simultaneously with the training of the model. Hence, Lasso has been widely applied, and its theoretical properties have been extensively studied (Zou, 2006; Hastie et al., 2015). On the other hand, Lasso is basically a feature selection method limited to linear models, and there is no guarantee that it can capture nonlinear relationships.

A widely used approach for accurately capturing nonlinear relationships is the maximum relevance (MR) feature selection, which introduces a statistical score that evaluates the nonlinear relationship between each feature and the target variable and selects the top features with the greatest relevance to the output (Peng et al., 2005). Generally, the mutual information (Cover and Thomas, 2006), distance correlation (Székely and Rizzo, 2009), and the Hilbert-Schmidt independence criterion (HSIC) (Gretton et al., 2005) are often used as scores. Although MR feature selection methods are efficient and can be applied to high-dimensional and large sample problems without difficulty, they are apt to select redundant features because they only use input-output relevance and not input-input relevance. To overcome this, the minimum redundant maximum relevance (mRMR) feature selection can select a subset of non-redundant features that are highly relevant to the output (Peng et al., 2005). Although mRMR has been experimentally shown to be superior to MR feature selection methods, three disadvantages have been pointed out mainly: the op-

Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151. Copyright 2022 by the author(s).

timization problem is discrete, this problem must be solved using a greedy approach, and the mutual information estimation is difficult (Walters-Williams and Li, 2009; Climente-González et al., 2019).

In contrast, Yamada et al. (2014) proposed a kernel-based mRMR method, called HSIC Lasso. HSIC Lasso adopts HSIC instead of the mutual information as the score to measure the dependence between variables and can select a subset of features using an L_1 penalty term. HSIC Lasso is a convex optimization problem, which can find the globally optimal solution and has been empirically reported to be superior to most feature selection methods. The Sparse HSIC (SpHSIC) regression framework that included HSIC Lasso can be regarded as a continuous optimization variant of mRMR, and more recently, its asymptotic theory has been established (Poignard and Yamada, 2020). Furthermore, some application methods based on HSIC Lasso have also been proposed, such as unsupervised nonlinear feature selection methods (Huang et la., 2020) and local interpretable model explanation methods for Graph Neural Networks (Huang et la., 2020). Of course, focusing on HSIC as an independence criterion, for example, HSIC-based post-selection inference (PSI) algorithm that can find a set of statistically significant features from non-linearly related data (Yamada et al., 2018) and PSI framework for divergence measure, which can introduces a general hypothesis test for PSI and select a set of statistically significant features that discriminate two distributions (Yamada et al., 2018), have been proposed.

HSIC Lasso is very attractive, and to the best of our knowledge, is currently one of the most effective feature selection methods. However, although the empirically obtained solutions are sparse, the number of non-zero solutions is rather large compared with the true optimal number of features, and this tends to lead to the adoption of a strategy of selecting from the top in order with a predefined number of selections, as with many other nonlinear feature selection methods. In certain problem settings, the boundary between relevant and irrelevant features is ambiguous. Overcoming this problem is the main motivation for this study.

In this paper, we propose an extension of HSIC Lasso for selecting important nonlinear features more clearly by setting weights for the coefficients that evaluate the association between each feature and the target variable and adaptively adjusting these weights from the data on the basis of a stochastic model. Introducing a prior distribution that induces sparsity generally breaks the conjugate property with the likelihood. Moreover, when the non-negative constraints on the coefficients specific to HSIC Lasso are also incorporated into the model in the form of a truncated prior distribution, inference becomes even more expensive. Our main contribution is that the proposed method does not use a truncated prior but directly add non-negative constraints to the closed-form update rules derived from variational inference, which allows us to select nonlinear features more clearly and effectively. In addition, the proposed method is simple to implement, which is highly preferable for practitioners. Furthermore, various innovations proposed as derivatives of HSIC Lasso (Yamada et al., 2018; Climente-González et al., 2019) can be integrated into the proposed method, and scalability to high-dimensional and large datasets is higher than HSIC Lasso, experimentally. Experiments on synthetic and real-world datasets showed that the proposed method is promising, and in certain problem settings, it assigned non-zero importance only on the correct set of the relevant features in model selection by maximizing the marginal likelihood.

2 BACKGROUND

In this section, we briefly review the current feature selection methods related to the proposed method.

2.1 Problem Description

Let $\mathcal{X} \subset \mathbb{R}^P$ be the domain of input vector $\boldsymbol{x}$ and $\mathcal{Y} \subset \mathbb{R}$ be the domain of output sample y. Suppose we are given N independent and identically distributed (i.i.d.) paired samples $\mathcal{D} = \{(\boldsymbol{x}^{(n)}, y^{(n)}) \in \mathcal{X} \times \mathcal{Y} | n = 1, \ldots, N\}$ drawn from a joint distribution. Then, with the p-th feature as $\boldsymbol{x}_p \in \mathbb{R}^N$, we denote the original input data as $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_P] \in \mathbb{R}^{N \times P}$. The purpose of supervised feature selection is to find as few M features ($M < P$) as possible in the input data $\boldsymbol{X}$ that are involved in predicting the output vector $\boldsymbol{y}$.

2.2 mRMR

Let $\mathrm{Sc}(\cdot, \cdot) \geq 0$ be an association score between two variables. Using this, the objective function of the ordinary mRMR feature selection method can be reformulated as

$$\begin{aligned} &\arg\max_{\mathcal{I}} \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \mathrm{Sc}(\boldsymbol{x}_i, \boldsymbol{y}) - \frac{1}{|\mathcal{I}|^2} \sum_{i,j \in \mathcal{I}} \mathrm{Sc}(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ =&\arg\max_{\boldsymbol{\beta} \in \{0,1\}^P} \sum_{p=1}^{P} \frac{\beta_p}{\|\boldsymbol{\beta}\|_1} \mathrm{Sc}(\boldsymbol{x}_p, \boldsymbol{y}) - \sum_{p,p'=1}^{P} \frac{\beta_p \beta_{p'}}{\|\boldsymbol{\beta}\|_1^2} \mathrm{Sc}(\boldsymbol{x}_p, \boldsymbol{x}_{p'}) \end{aligned} \tag{1}$$

where $\mathcal{I}$ is the set of selected feature indices, and $\boldsymbol{\beta}$ is the binary vector based on $\mathcal{I}$ (Peng et al., 2005). Note that when $\mathrm{Sc}(\cdot, \cdot)$ takes large negative values if the selected features are not mutually independent, the first term in (1) selects features that are non-independent of the output vector $\boldsymbol{y}$, and the second term is the penalized term that selects independent features. Thus, by selecting features by maximizing (1), we can select features that are dependent on the output, and the selected features are mutually independent. For the association score Sc, HSIC can be used as well as the mutual information commonly seen.

HSIC is a kernel-based independence measure (Gretton et al., 2005). For N-dimensional sample vector $\boldsymbol{z} \in \mathbb{R}^N$, we denote the kernel matrix as $[\boldsymbol{K_z}]_{ij} = k(z^{(i)}, z^{(j)})$, with $k(z, z')$ as the kernel function. Furthermore, let $\bar{\boldsymbol{K}}_{\boldsymbol{z}} = \boldsymbol{\Gamma}_N \boldsymbol{K_z} \boldsymbol{\Gamma}_N$ be the centered kernel matrix, where $\boldsymbol{\Gamma}_N = \boldsymbol{I}_N - \frac{1}{N}\boldsymbol{1}_N\boldsymbol{1}_N^\top$ is the centering matrix, $\boldsymbol{I}_N$ is the N dimensional identity matrix, and $\boldsymbol{1}_N$ is the N dimensional vector with all ones. Using such a centered Gram matrix, the HSIC estimator between two sample vectors $\boldsymbol{z}_p, \boldsymbol{z}_{p'} \in \mathbb{R}^N$, can be simply expressed as $\mathrm{HSIC}(\boldsymbol{z}_p, \boldsymbol{z}_{p'}) = \mathrm{Tr}(\bar{\boldsymbol{K}}_{\boldsymbol{z}_p}\bar{\boldsymbol{K}}_{\boldsymbol{z}_{p'}})$.

The normalized cross-covariance operator (NOCCO) is also important as a kernel-based dependence measure related to HSIC (Fukumizu et al., 2008). Using $\tilde{\boldsymbol{K}}_{\boldsymbol{z}} = \bar{\boldsymbol{K}}_{\boldsymbol{z}}(\bar{\boldsymbol{K}}_{\boldsymbol{z}} + \epsilon N \boldsymbol{I}_N)^{-1}$ instead of $\bar{\boldsymbol{K}}_{\boldsymbol{z}}$, NOCCO is formulated as $\mathrm{NOCCO}(\boldsymbol{z}_p, \boldsymbol{z}_{p'}) = \mathrm{Tr}(\tilde{\boldsymbol{K}}_{\boldsymbol{z}_p}\tilde{\boldsymbol{K}}_{\boldsymbol{z}_{p'}})$, similar to HSIC, where $\epsilon > 0$ is the regularization parameter. Because NOCCO was shown to be asymptotically independent of the choice of kernels, it is expected to be less sensitive to the kernel parameter choice than HSIC (Yamada et al., 2014).

2.3 HSIC Lasso

Note that $\boldsymbol{\beta}$ is a sparse binary vector; thus, we can obtain the following optimization problem from (1) relaxed by $\boldsymbol{\omega} \in \mathbb{R}_+^P$, using HSIC as the association score,

$$\underset{\boldsymbol{\omega}\in\mathbb{R}_+^P}{\arg\max} \sum_{p=1}^{P} \omega_p \mathrm{HSIC}(\boldsymbol{x}_p, \boldsymbol{y}) - \sum_{p,p'=1}^{P} \omega_p \omega_{p'} \mathrm{HSIC}(\boldsymbol{x}_p, \boldsymbol{x}_{p'}) - \lambda \|\boldsymbol{\omega}\|_1 \qquad (2)$$

where $\lambda \geq 0$ is the regularization parameter. The non-negative constraint is then added to $\boldsymbol{\omega}$ because the original $\boldsymbol{\beta}$ is non-negative. Yamada et al. (2014) focused on the fact that (2) is a convex optimization problem and proposed the following HSIC Lasso as an equivalent optimization problem that can handle high-dimensional problems,

$$\min_{\boldsymbol{\omega}\in\mathbb{R}_+^P} \frac{1}{2} \left\| \bar{\boldsymbol{K}}_{\boldsymbol{y}} - \sum_{p=1}^{P} \omega_p \bar{\boldsymbol{K}}_{\boldsymbol{x}_p} \right\|_F^2 + \lambda \|\boldsymbol{\omega}\|_1. \qquad (3)$$

For the specific optimization of (3), the dual augmented Lagrangian (DAL) algorithm (Tomioka et al., 2011) can be computationally highly efficient and incorporate the non-negative constraint without losing its computational advantages. Moreover, an efficient Least Angle Regression (LARS) (Efron et al., 2004) based method has been proposed that can scale up HSIC Lasso to handle high-dimensional and large-scale datasets (Yamada et al., 2018). Additionally, Block HSIC Lasso, which greatly reduces the memory complexity of the kernel matrices, by splitting the data in blocks, and using the block HSIC estimator (Zhang et al., 2018) to estimate the HSIC terms has also been proposed (Climente-González et al., 2019). Recently, the SpHSIC regression framework, which includes the HSIC Lasso, has been proposed. In the framework, theoretical considerations such as the oracle property have been made. (Poignard and Yamada, 2020).

Note that the NOCCO introduced in subsection 2.2 can be formulated as NOCCO Lasso by using $\tilde{\boldsymbol{K}}$ instead of $\bar{\boldsymbol{K}}$ in (3) (Yamada et al., 2014). The extensions to HSIC Lasso introduced in this subsection can be applied to NOCCO Lasso.

3 PROPOSED METHOD

In this section, we formally describe our stochastic model and a suitable approximate inference scheme.

3.1 Key Ideas

For the first key idea, we weight the parameter $\boldsymbol{\omega}$ in (3) with $\boldsymbol{\eta} \in \mathbb{R}_+^P$, aiming for a more distinct nonlinear feature selection than with HSIC Lasso. In this process, we introduce a stochastic model based on the HSIC Lasso optimization problem and adaptively infer the optimal weights $\boldsymbol{\eta}$ from paired samples $\mathcal{D}$. The second key idea is to derive approximate closed-form update equations from variational inference and add HSIC-Lasso-specific non-negative constraints directly to these update equations. The family of super-Gaussian distributions is assumed as a prior distribution to induce sparsity in the proposed method. Dealing with such a distribution as a truncated prior defined only in $\mathbb{R}_+$, sampling methods with high computational complexity or approximate inference is needed to to learn parameters. We confirmed that the proposed method avoids these problems and achieves very high feature selection performance with shorter execution time than HSIC Lasso.

3.2 Problem Formulation

Let $\bar{\boldsymbol{\kappa}} \in \mathbb{R}^{N^2}$ be a vector of kernel matrix $\bar{\boldsymbol{K}}$ appropriately flattened and $\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}} = [\bar{\boldsymbol{\kappa}}_{\boldsymbol{x}_1}, \ldots, \bar{\boldsymbol{\kappa}}_{\boldsymbol{x}_P}] \in \mathbb{R}^{N^2 \times P}$ be a matrix of all flat vectors for features. Then, the loss term of HSIC Lasso in (3) can be rewritten as follows:

$$\left\| \bar{\boldsymbol{K}}_{\boldsymbol{y}} - \sum_{p=1}^{P} \omega_p \bar{\boldsymbol{K}}_{\boldsymbol{x}_p} \right\|_F^2 = \|\bar{\boldsymbol{\kappa}}_{\boldsymbol{y}} - \bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}\boldsymbol{\omega}\|_2^2 . \tag{4}$$

On the basis of (4), we assume the classical Gaussian linear model with i.i.d. noise with variance σ^2, that is,

$$\bar{\boldsymbol{\kappa}}_{\boldsymbol{y}} \mid \boldsymbol{\omega} \sim \mathcal{N}(\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}\boldsymbol{\omega}, \sigma^2 \boldsymbol{I}). \tag{5}$$

We also assume that $\{\omega_p\}_{p=1}^{P}$ are jointly independent, and each ω_p has a symmetric density with inverse scale parameter η_p as follows:

$$p(\boldsymbol{\omega}|\boldsymbol{\eta}) = \prod_{p=1}^{P} p(\omega_p|\eta_p) = \prod_{p=1}^{P} q(|\omega_p|\eta_p^{\frac{1}{2}})\eta_p^{\frac{1}{2}}, \tag{6}$$

where q is a distribution inducing sparsity. This is a common assumption used in Bayesian Lasso (Park and Casella, 2008), and in addition, correlations tend to be eliminated by the mRMR framework. Given samples $\mathcal{D}$, our aim is to infer the parameters, including $\boldsymbol{\eta}$, by maximizing the marginal likelihood $p(\bar{\boldsymbol{\kappa}}_{\boldsymbol{y}}|\boldsymbol{\eta})$.

3.3 Super-Gaussian Priors

In this paper, we use scale mixtures of Gaussians prior as the family of distributions that induce sparsity. Let q be a Gaussian scale mixture for a mixing density $r(t)$, that is, $q(u) = \int_0^\infty \mathcal{N}(u|0,t)r(t)dt$. This formulation includes the Laplace distribution and Student ’ s t, which are often used in Bayesian Lasso due to their suitability for modeling sparsity (Park and Casella, 2008; Shervashidze and Bach, 2015; Van Erp et al., 2019). For learning such models, we introduce an approximate Gaussian posterior and derive variational optimization with closed-form update equations.

Focusing on the fact that q is also super-Gaussian, $\log q(u)$ is convex and non-increasing in u^2. Thus, according to previous study (Palmer et al., 2006), we can obtain representation of the following form by convex conjugacy,

$$\log q(u) = \sup_{s \in \mathbb{R}_+} \left\{ -\frac{u^2}{2s} - \phi(s) \right\}, \tag{7}$$

where $\phi(s)$ is convex in $1/s$ and the expression inside the supremum in (7) has a unique maximizer. Consequently, we obtain the following variational representation for $p(\omega_p|\eta_p)$ by combining (6) and (7),

$$p(\omega_p|\eta_p) = \eta_p^{\frac{1}{2}} \sup_{s_p \in \mathbb{R}_+} \left\{ \mathcal{N}\left(\omega_p \middle| 0, \frac{s_p}{\eta_p}\right) \left(\frac{2\pi s_p}{\eta_p}\right)^{\frac{1}{2}} \exp(-\phi(s_p)) \right\}. \tag{8}$$

3.4 Variational Inference

This stochastic model described above, including the combination of the likelihood (5) and the variational representation of the prior (8), leads to the following variational bound on the marginal distribution,

$$\begin{aligned}
&\log\, p(\bar{\boldsymbol{\kappa}}_{\boldsymbol{y}}|\boldsymbol{\eta}) = \log \int_{\mathbb{R}^P} \mathcal{N}\left(\bar{\boldsymbol{\kappa}}_{\boldsymbol{y}} \middle| \bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}\boldsymbol{\omega}, \sigma^2 \boldsymbol{I}\right) \prod_{p=1}^{P} p(\omega_p|\eta_p) d\omega_p \\
&\geq \sup_{\boldsymbol{s} \in \mathbb{R}_+^P} \Bigg[\log \mathcal{N}\left(\bar{\boldsymbol{\kappa}}_{\boldsymbol{y}} \middle| \boldsymbol{0}, \bar{\boldsymbol{\kappa}}_{\boldsymbol{X}} \boldsymbol{\Xi}^{-1} \bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}^\top + \sigma^2 \boldsymbol{I}\right) \\
&+ \sum_{p=1}^{P} \left\{ \frac{1}{2}\log \eta_p + \frac{1}{2}\log(2\pi) + \frac{1}{2}\log\left(\frac{s_p}{\eta_p}\right) - \phi(s_p) \right\} \Bigg] \\
&= -\inf_{\boldsymbol{\mu} \in \mathbb{R}^P} \inf_{\boldsymbol{\Sigma} \succeq 0} \inf_{\boldsymbol{s} \in \mathbb{R}_+^P} \Bigg[\frac{1}{2\sigma^2} \|\bar{\boldsymbol{\kappa}}_{\boldsymbol{y}} - \bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}\boldsymbol{\mu}\|_2^2 + \frac{1}{2}\boldsymbol{\mu}^\top \boldsymbol{\Xi} \boldsymbol{\mu} \\
&+ \frac{1}{2\sigma^2} \mathrm{Tr}\left(\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}^\top \bar{\boldsymbol{\kappa}}_{\boldsymbol{X}} \boldsymbol{\Sigma}\right) + \frac{1}{2}\mathrm{Tr}\left(\boldsymbol{\Xi}\boldsymbol{\Sigma}\right) - \frac{1}{2}\log\det \boldsymbol{\Sigma} \\
&+ \sum_{p=1}^{P} \left\{ \phi(s_p) - \frac{1}{2}\log \eta_p \right\} + \frac{N^2}{2}\log(2\pi\sigma^2) - \frac{P}{2}\log(2\pi \mathrm{e}) \Bigg] \\
&=: -\inf_{\boldsymbol{\mu} \in \mathbb{R}^P} \inf_{\boldsymbol{\Sigma} \succeq 0} \inf_{\boldsymbol{s} \in \mathbb{R}_+^P} f(\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{s}),
\end{aligned} \tag{9}$$

where $\boldsymbol{\Xi} \in \mathbb{R}^{P \times P}$ is a diagonal matrix such that $[\boldsymbol{\Xi}]_{pp} = \eta_p / s_p$. Note that $\boldsymbol{\mu} \in \mathbb{R}^P$ and $\boldsymbol{\Sigma} \in \mathbb{R}^{P \times P}$ respectively correspond to the mean vector and covariance matrix of the posterior $\mathcal{N}(\boldsymbol{\omega}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$, when $\mathcal{N}(\bar{\boldsymbol{\kappa}}_{\boldsymbol{y}}|\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}\boldsymbol{\omega}, \sigma^2\boldsymbol{I})$ is a likelihood and $\mathcal{N}(\boldsymbol{\omega}|\boldsymbol{0}, \boldsymbol{\Xi}^{-1})$ is a prior. Since we assume that the final optimal solution for $\boldsymbol{\omega}$ is obtained by maximum a posteriori estimation, the proposed method treats the optimal solution for $\boldsymbol{\mu}$ as the optimal solution for $\boldsymbol{\omega}$.

The reason for not using a truncated prior is that the first integral in (9) to derive the marginal likelihood becomes $\mathbb{R}^P \to \mathbb{R}_+^P$. As described in subsection 3.1, because introducing a prior distribution that induces sparsity generally breaks the conjugate property with the likelihood, it is very difficult to solve this integral analytically. The non-negative constraint of the proposed method is an idea to avoid computational cost.

3.5 Non-negative Constraint

With the proposed method, instead of using a truncated prior distribution, we directly add a non-negative constraint with the following penalty term for $\boldsymbol{\mu}$ in (9),

$$\psi(\boldsymbol{\mu}) := \sum_{p=1}^{P} \psi'(\mu_p) := \sum_{p=1}^{P} \left\{ \begin{array}{ll} \infty & (\mu_p < 0) \\ \lambda\mu_p & (\mu_p \geq 0) \end{array} \right. , \tag{10}$$

where $\lambda > 0$ is the regularization parameter. Thus, since $\psi(\boldsymbol{\mu}) \geq 0$ for any $\boldsymbol{\mu} \in \mathbb{R}^P$, $\inf\,(f(\boldsymbol{\mu}) + \psi(\boldsymbol{\mu})) \geq \inf f(\boldsymbol{\mu})$ is satisfied by the infimum properties. Therefore, the marginal distribution (9) can be set to the following novel variational lower bound with non-negative constraints.

$$\log p(\bar{\boldsymbol{\kappa}}_{\boldsymbol{y}}|\boldsymbol{\eta}) \geq -\inf_{\boldsymbol{\mu}\in\mathbb{R}^P}\inf_{\boldsymbol{\Sigma}\succeq 0}\inf_{\boldsymbol{s}\in\mathbb{R}_+^P}\{f(\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{s})+\psi(\boldsymbol{\mu})\}$$
$$= -\inf_{\boldsymbol{\mu}\in\mathbb{R}_+^P}\inf_{\boldsymbol{\Sigma}\succeq 0}\inf_{\boldsymbol{s}\in\mathbb{R}_+^P}\{f(\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{s})+\lambda\|\boldsymbol{\mu}\|_1\} \tag{11}$$

3.6 Update Equations

Consequently, as an approximation to maximizing the log-likelihood, we consider optimizing the right side of (11). However, in the form given by (11), the variational lower bound is difficult to optimize. To overcome this problem, we regard parts of it as minima of convex functions, design an iterative algorithm with analytic updates, and find a local minimum. As a result, the optimization problem to be solved with the proposed method, which also includes $\boldsymbol{\eta}$ and σ^2, is equivalent to iterating the following closed-form update equations until convergence.

$$\boldsymbol{\mu} \leftarrow \arg\inf_{\boldsymbol{\mu}\in\mathbb{R}_+^P}\left\{\frac{1}{2\sigma^2}\|\bar{\boldsymbol{\kappa}}_{\boldsymbol{y}}-\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}\boldsymbol{\mu}\|_2^2+\frac{1}{2}\boldsymbol{\mu}^\top\boldsymbol{\Xi}\boldsymbol{\mu}+\lambda\|\boldsymbol{\mu}\|_1\right\} \tag{12}$$

$$\boldsymbol{\Sigma} \leftarrow \sigma^2\left(\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}^\top\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}+\sigma^2\boldsymbol{\Xi}\right)^{-1} \tag{13}$$

$$\boldsymbol{s} \leftarrow \arg\inf_{\boldsymbol{s}\in\mathbb{R}_+^P}\left\{\frac{1}{2}\boldsymbol{\mu}^\top\boldsymbol{\Xi}\boldsymbol{\mu}+\frac{1}{2}\mathrm{Tr}\left(\boldsymbol{\Xi}\boldsymbol{\Sigma}\right)+\sum_{p=1}^{P}\phi(s_p)\right\} \tag{14}$$

$$\boldsymbol{\eta} \leftarrow \boldsymbol{s}\oslash(\boldsymbol{\mu}\odot\boldsymbol{\mu}+\mathrm{diag}\boldsymbol{\Sigma}) \tag{15}$$

$$\sigma^2 \leftarrow \frac{1}{N^2}\left(\|\bar{\boldsymbol{\kappa}}_{\boldsymbol{y}}-\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}\boldsymbol{\mu}\|_2^2+\mathrm{Tr}\left(\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}^\top\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}\boldsymbol{\Sigma}\right)\right) \tag{16}$$

where $\odot$ and $\oslash$ are the Hadamard product and Hadamard division, respectively, and $\mathrm{diag}\boldsymbol{\Sigma}$ is a vector formed from the diagonal of $\boldsymbol{\Sigma}$. Assuming the noise level known, we can also choose not to include (16) in the iteration. We finally regard the convergence value of $\boldsymbol{\mu}$ as the optimal $\boldsymbol{\omega}$.

Even with these update equations, the proposed method also holds the advantages of HSIC Lasso. In particular, recent extension techniques are equally applicable to the proposed method. For example, the efficient search method by LARS (Yamada et al., 2018) can be applied to (12) as well, and the block HSIC estimator (Climente-González et al., 2019) can be applied to $\bar{\boldsymbol{K}}_y$ and $\bar{\boldsymbol{K}}_{x_p}$ of the proposed method as well because of the approximation of the kernel matrices.

3.7 Prior: Generalized Gaussian

The family of super-Gaussian distributions includes Generalized Gaussian. The density of this distribution is given by

$$p(\omega_p|\eta_p,\alpha,\beta) = \frac{\eta_p^{\frac{1}{2}}\beta}{2\alpha\Gamma\left(\frac{1}{\beta}\right)}\exp\left(-\left(\frac{|\omega_p|\eta_p^{\frac{1}{2}}}{\alpha}\right)^{\beta}\right) \tag{17}$$

where $\alpha \in \mathbb{R}_+$ and $\beta \in \mathbb{R}_+$ are the scale and shape parameters respectively, can be also rewritten in this variational representation for $0 < \beta \leq 2$. For this distribution,

$$\phi(s_p) = \left\{\frac{1}{\alpha^\beta}\left(\frac{2\alpha^\beta}{\beta}\right)^{\frac{\beta}{\beta-2}}-\left(\frac{2\alpha^\beta}{\beta}\right)^{\frac{2}{\beta-2}}\right\}(2s_p)^{-\frac{\beta}{\beta-2}}$$
$$-\log\beta+\log 2\alpha\Gamma\left(\frac{1}{\beta}\right). \tag{18}$$

In particular, the Laplace distribution, which is often used in Bayesian Lasso (Park and Casella, 2008), corresponds to $\beta = 1$, and $\phi(s_p)$ can be simply rewritten as

$$\phi(s_p) = \frac{s_p}{2\alpha^2}+\log 2\alpha. \tag{19}$$

Therefore, the update equation (14) for $\boldsymbol{s}$ is rewritten as

$$\boldsymbol{s} \leftarrow \alpha\sqrt{\boldsymbol{\eta}\odot(\boldsymbol{\mu}\odot\boldsymbol{\mu}+\mathrm{diag}\boldsymbol{\Sigma})}. \tag{20}$$

3.8 Prior: Student's t

Recently, Student's t can be used as well as the Laplace distribution (Van Erp et al., 2019). The density of this distribution is given by

$$p(\omega_p|\eta_p,\nu) = \left(\frac{\eta_p}{2\pi}\right)^{\frac{1}{2}}\frac{\Gamma\left(\nu+\frac{1}{2}\right)}{\Gamma\left(\nu\right)}\left(1+\frac{\eta_p\omega_p^2}{2}\right)^{-\nu-\frac{1}{2}} \tag{21}$$

where ν is a shape parameter. The smaller the ν, the heavier-tailed the distribution (note that there is no finite variance for $\nu \leq 1$). For this distribution,

$$\phi(s_p) = \frac{1}{s_p}+\left(\nu+\frac{1}{2}\right)\log\left(s_p\right)+\mathrm{const.} \tag{22}$$

Therefore, the update equation (14) for $\boldsymbol{s}$ is rewritten as

$$\boldsymbol{s} \leftarrow \frac{1}{\nu+\frac{1}{2}}\left\{1+\frac{\boldsymbol{\eta}}{2}\odot(\boldsymbol{\mu}\odot\boldsymbol{\mu}+\mathrm{diag}\boldsymbol{\Sigma})\right\}. \tag{23}$$

3.9 Computational Complexity

We first need $\mathcal{O}(N^2P)$ to construct the kernel matrices from the data $\mathcal{D}$. After this calculation, the update equations (12) and (13) have the highest computational complexity. Equation (12), which is an optimization problem similar to Elastic-Net (Zou and

Hastie, 2005) since Ξ is a diagonal matrix, can be easily solved using well-known algorithms such as DAL and LARS with similar uses as HSIC Lasso (Yamada et al., 2014; Yamada et al., 2018). Then, as well as HSIC Lasso, we can dramatically reduce the computational complexity of the kernel matrices and (12) with previously proposed methods (Yamada et al., 2018; Climente-González et al., 2019). For (13), while we need $\mathcal{O}(P^3)$ to compute the inverse matrix, since $\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}^\top \bar{\boldsymbol{\kappa}}_{\boldsymbol{X}} + \sigma^2 \Xi$ is always a positive definite matrix, we can compute relatively fast with the Cholesky decomposition. Empirically, compared with HSIC Lasso, the proposed method is able to calculate significantly faster in both $N \ll P$ and $N \gg P$. This experiment is shown in Experiments 4.2 (see Figures 1 (f)).

4 EXPERIMENTS

In this section, we discuss experimentally investigating the performance of the proposed and other feature selection methods using synthetic and real-world datasets.

4.1 Set-up

On the basis of the results of several previous studies (Gretton et al., 2005; Song et al., 2012), we adopt the following universal kernel function for all of our experiments. For input kernels, we first normalize the input features to have unit standard deviation then use the Gaussian kernel:

$$k(z, z') = \exp\left(-\frac{(z - z')^2}{2\sigma_{\boldsymbol{z}}^2}\right) \tag{24}$$

where $\sigma_{\boldsymbol{z}}^2$ is the kernel width, and we fix $\sigma_{\boldsymbol{z}} = 1$. For output kernels in regression cases (i.e., $\boldsymbol{y} \in \mathbb{R}^N$), we similarly normalize $\boldsymbol{y}$ to have unit standard deviation then use the same Gaussian kernel. By contrast, for output kernels in classification cases (i.e., $\boldsymbol{y}$ is categorical), we use the delta kernel:

$$k(z, z') = \begin{cases} 1/n_z & \text{if } z = z' \\ 0 & \text{otherwise} \end{cases} \tag{25}$$

where n_z is the number of observations in class z.

As one of the metrics, we use the redundancy rate (RED) (Zhao et al., 2010) to check whether a method can successfully select non-redundant features:

$$\mathrm{RED}(\boldsymbol{X}_{\mathcal{I}}) = \frac{1}{|\mathcal{I}|(|\mathcal{I}| - 1)} \sum_{i,j \in \mathcal{I},\, i \neq j} |\rho_{i,j}| \tag{26}$$

where $\mathcal{I}$ is the index set of the selected features, and $\rho_{i,j}$ is the Pearson correlation coefficient between the i-th and j-th features. A large RED indicates that selected features are more strongly correlated with each other, that is, many redundant features are selected. Therefore, a small redundancy rate is preferable for the feature selection methods.

We compared the proposed method applying the HSIC or NOCCO kernel matrices with HSIC Lasso and NOCCO Lasso (Yamada et al., 2014; Yamada et al., 2018; Climente-González et al., 2019). The implementation is based on the following Github[1]. For all experiments, we use Student ' s t as a prior distribution and set $\nu = 1.5$ for the proposed method and $\epsilon = 10^{-3}$ in the NOCCO kernel matrices, and then, we randomly initialize the other parameters based on random seeds. We also used an Ubuntu 18.04 server with 96-core Intel Xeon Platinum 2.7 GHz and 1.5 TB RAM memory.

4.2 Synthetic Datasets

First, we considered a regression problem from a high-dimensional input to verify the performance of the proposed method using the following two synthetic datasets. For comparison, we used the data generated from the same method as that used in a previous study (Yamada et al., 2014).

Data1: Additive model

$$y = -2\sin(2x_1) + x_2^2 + x_3 + \exp(-x_4) + \epsilon \tag{27}$$

where $[x_1, \ldots, x_{256}]^\top \sim \mathcal{N}(\mathbf{0}_{256}, \boldsymbol{I}_{256})$ and $\epsilon \sim \mathcal{N}(0, 1)$.

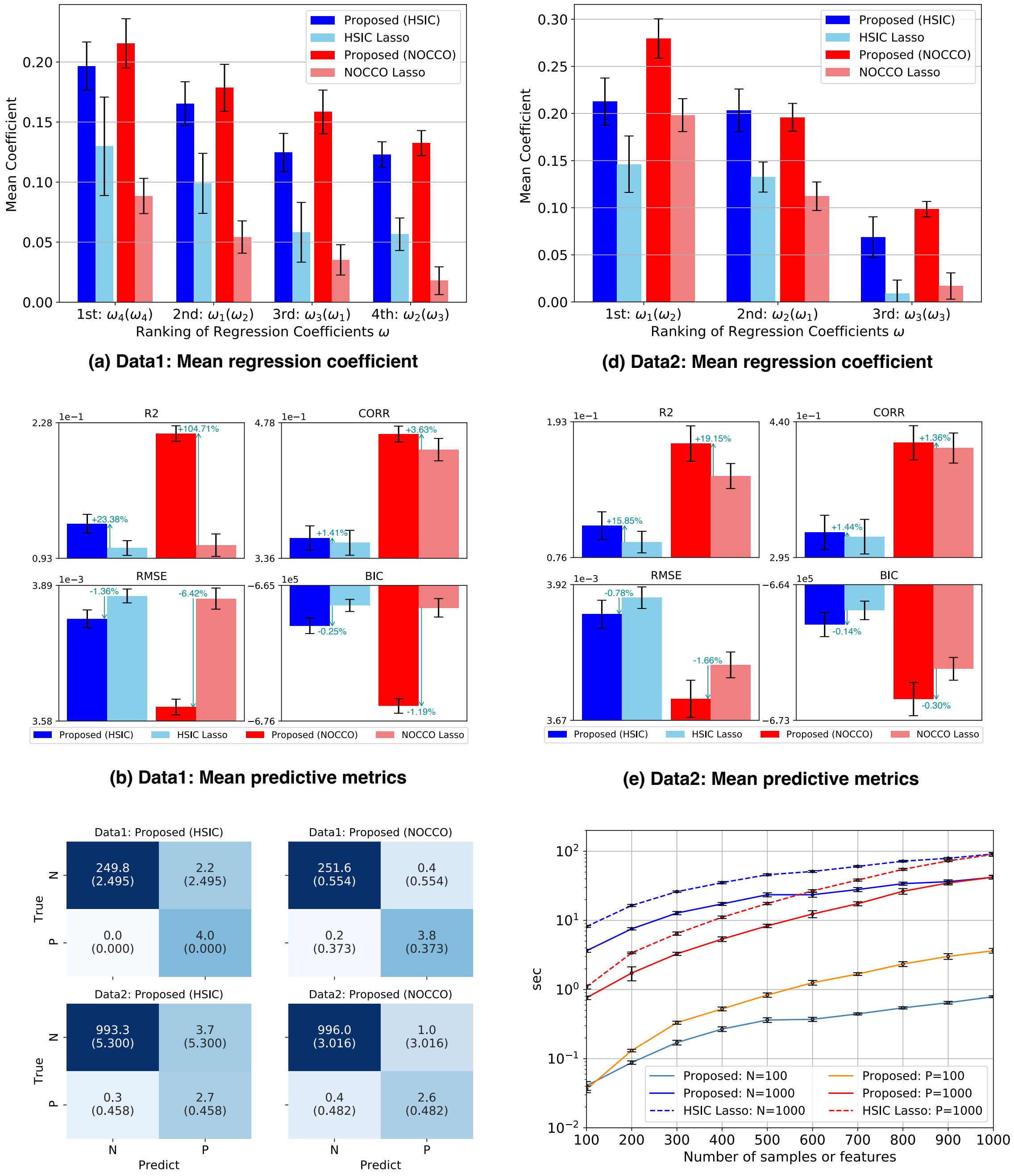
Data2: Non-additive model

$$y = x_1 \exp(2x_2) + x_3^3 + \epsilon \tag{28}$$

where $[x_1, \ldots, x_{1000}]^\top \sim \mathcal{N}(\mathbf{0}_{1000}, \boldsymbol{I}_{1000})$ and $\epsilon \sim \mathcal{N}(0, 1)$.

As shown in Figures 1 (a) and (d), we set $N = 1000$ and λ so that the number of non-zero coefficients would be M^* for each method over 30 runs, where M^* is the number of true features (i.e. $M^* = 4$ for Data1 and $M^* = 3$ for Data2). To detect important features easily, using a threshold for example, it is desirable that non-zero coefficients of important features are sufficiently larger than those of non-important features. In order to make a fair comparison from this perspective, we treat M^* as a known constant in these experiments. Clearly, the proposed method had more distinct regression coefficients than HSIC Lasso for both HSIC and NOCCO. This is because the proposed method adaptively adjusts the weights of the regression coefficients on the basis of the stochastic model from the data, which robustly keeps the important features

[1] https://github.com/riken-aip/pyHSICLasso



(a) Data1: Mean regression coefficient

(d) Data2: Mean regression coefficient

(b) Data1: Mean predictive metrics

(e) Data2: Mean predictive metrics

(c) Data1 & Data2: Mean confusion matrix

(f) Data2: Computation time

Figure 1: Results for two synthetic datasets. (a),(d): Mean regression coefficients of the top M^* for $N = 1000$ over 30 runs. Horizontal axis denotes top M^* features for HSIC (NOCCO in brackets), and vertical axis denotes coefficient value. Regularization parameter λ is set so that number of non-zero coefficients is M^* in each method. (b),(e): Mean predictive metrics for the test data over 30 runs. These metrics are based on the loss function (4). R2, CORR, RMSE and BIC mean coefficient of determination, correlation coefficient, root mean square error, and bayesian information criterion, respectively. Moreover, The green arrows and numbers mean the improvement rate of the proposed method over HSIC (NOCCO) Lasso. Note that R2 and CORR indicate the rate of increase and RMSE and BIC indicate the rate of decrease. (c): Average of confusion matrices with proposed method over 30 runs. We assign binary labels with (P) relevant and (N) irrelevant features, and in prediction we consider non-zero coefficients to be relevant. Regularization parameter λ is set so that lower bound of marginal likelihood (9) is maximized. (f): Comparison of computation time for Data2. Horizontal axis denotes number of training samples or features, and vertical axis denotes computation time in log-scale.

Table 1: Mean classification scores and RED values for Gas Sensor dataset. AC, BAC, and F1 mean classification accuracy, balanced classification accuracy, and F1 score, respectively.

	AC	**BAC**	**F1**	**RED**
Proposed (HSIC)	**0.978**	**0.975**	**0.978**	0.253
Proposed (NOCCO)	0.977	0.974	0.977	0.275
HSIC Lasso	0.972	0.967	0.972	0.242
NOCCO Lasso	0.972	0.967	0.972	0.235
FsNet ($H = 20$)	0.846	0.849	0.846	0.093
FsNet ($H = 40$)	0.843	0.847	0.843	0.091
FsNet ($H = 60$)	0.842	0.847	0.842	0.092
FsNet ($H = 80$)	0.834	0.838	0.834	**0.089**

against the increase in λ. Additionally, in this case, the proposed method also improves the regression metrics for the test data, as shown in Figures 1 (b) and (e).

Moreover, Figure 1 (c) shows the mean confusion matrix from the proposed method when λ is tuned by maximizing the marginal likelihood (9) over 30 runs. We classified each feature with binary labels of relevant features (P) and irrelevant features (N), and considered non-zero regression coefficients in the predictive labels as relevant features. In particular, the accuracy was perfect for 15 out of 30 runs. This suggests that the proposed method can effectively perform model selection with the marginal likelihood.

Finally, as shown in Figure 1 (f), we evaluated the computation time of the proposed method with respect to the number of samples and number of features used with Data2. For fairness, we randomly determined λ in each run. As a result, compared with HSIC Lasso, the computation time of the proposed method is significantly faster in both $N \ll P$ and $N \gg P$. Note that the results of HSIC Lasso are consistent with previous results (Yamada et al., 2014). While these results may depend on the convergence rate and computing environments, we consider that the proposed method is significantly faster than HSIC Lasso.

We consider the main reason for the advantages of the proposed method over HSIC Lasso is that the proposed method adopts a stochastic model and the parameter ω_p is multiplied by the weight η_p which can be adjusted by maximizing the marginal likelihood. The main advantages are: the power to detect important features is improved due to the larger difference between the non-zero and zero coefficients (Figures 1 (a) and (d)); the optimization based on the marginal likelihood can narrow down important features with high accuracy (Figure 1 (c)); the convergence and runtime is faster, probably due to the weights (Figure 1 (f)).

Table 2: Mean classification scores and RED values for USPS dataset. Note that abbreviations are the same as in Table 1.

	AC	**BAC**	**F1**	**RED**
Proposed (HSIC)	**0.960**	**0.955**	**0.960**	0.105
Proposed (NOCCO)	0.955	0.950	0.955	0.117
HSIC Lasso	0.957	0.953	0.957	0.119
NOCCO Lasso	0.948	0.942	0.948	0.135
FsNet ($H = 20$)	0.954	0.949	0.954	0.104
FsNet ($H = 40$)	0.955	0.950	0.955	0.104
FsNet ($H = 60$)	0.956	0.950	0.956	**0.103**
FsNet ($H = 80$)	0.955	0.950	0.955	0.104

4.3 Real-World Datasets

Next, we compared the performance of the same methods by using a multi-class classification task using real-world datasets. First of all, we would like to strongly argue that it is generally difficult to evaluate the performance in feature selection using real data, because important features are unknown. Therefore, a common practice is to input the selected features only into a machine learning algorithm that is irrelevant to feature selection, and to evaluate the performance of feature selection indirectly by its score.

In this paper, we used two real-world datasets, Gas Sensor[2] and USPS[3]. Gas Sensor contains chemical sensor data with 13910 instances, 128 features, and 6 classes, and USPS contains image data with 9298 instances, 256 features, and 10 classes.

For comparison in this experiment, we used DNN-based nonlinear feature selection method, called the feature selection network (FsNet), which comprises a selection layer that selects features and a reconstruction layer that stabilizes the training (Singh et al., 2020). The implementation is available at the following Github[4]. Here, we changed only the parameter "h_size" (= H), which specifies the number of nodes in the hidden layer in the code, and used the other parameters unchanged from the original.

In this experiment, we used 80% of the samples for training and the rest for testing. We repeated the experiment 30 times by randomly shuffling training and test samples, and λ was set so that the number of non-zero coefficients would be 50 (i.e. $M = 50$). We then used multi-class L_2-regularized kernel logistic regression (KLR) (Hastie et al., 2009) with the Gaussian kernel for evaluating classification scores of the selected features. In KLR, all tuning parameters are chosen based on 5-fold cross-validation.

[2] https://archive.ics.uci.edu/ml/datasets/

[3] https://jundongl.github.io/scikit-feature/

[4] https://github.com/singh-ml/fsnet

Table 1 and Table 2 show the mean classification scores and RED values for Gas Sensor dataset and USPS, respectively. For both datasets, the proposed method with HSIC achieved higher classification scores such as classification accuracy, balanced classification accuracy, and F1 score, and indicated highly effective in terms of classification performance. However, except for the results of FsNet, RED value of the proposed method was not significantly different from HSIC (NOCCO) Lasso. FsNet is reported to have performed better than HSIC Lasso for some datasets, but we were not able to confirm this in terms of classification accuracy for these data. Interestingly, RED value of FsNet is outstandingly low, which may suggest that reducing redundancy does not necessarily contribute to improving classification accuracy. Especially for USPS, the proposed method with HSIC achieved higher classification scores than HSIC (NOCCO) Lasso and the same level of low RED value as FsNet, simultaneously.

5 CONCLUSION

In this paper, we proposed an effective method for the HSIC Lasso framework, which use a stochastic model and variational inference. The regularization parameter tuned by the stochastic model-based marginal likelihood maximization of the proposed method can refine the relevant features with high accuracy in certain problem settings. We also emphasize that the coefficients of relevant features are relatively high with the proposed method, and the difference between relevant and irrelevant features is not ambiguous. This is because the proposed method adaptively adjusts the weights of the regression coefficients on the basis of the stochastic model from the data, which robustly keeps the important features against the increase in the regularization parameter.

For the future work, we will consider extending the proposed method to a hierarchical Bayesian model to carry out learning similar to Hierarchical Multiple Kernel Learning. Since such an extension can adopt multitask learning with different kernel functions, we expect to implement nonlinear feature selection that is more adaptive and with higher performance.

References

H. Abusamra (2013). A Comparative Study of Feature Selection and Classification Methods for Gene Expression Data of Glioma. *Procedia Computer Science*, 23:5–14.

H. Climente-González, C. Azencott, S. Kaski, and M. Yamada (2019). Block HSIC Lasso: model-free biomarker detection for ultra-high dimensional data. *Bioinformatics*, 35(14):i427–i435.

T. Cover and J. Thomas (2006). Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing). Wiley-Interscience.

B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani (2004). Least angle regression. *The Annals of statistics*, 32(2):407–499.

G. Forman (2008). BNS feature scaling: an improved representation over tf-idf for svm text classification. *In Proceedings of the 17th ACM conference on Information and knowledge management*, 263–270.

K. Fukumizu, A. Gretton, X. Sun, and B. Schölkopf (2008). Kernel measures of conditional dependence. *In Advances in neural information processing systems*, 489–496.

A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf (2005). Measuring statistical dependence with Hilbert-Schmidt norms. *In International conference on algorithmic learning theory*, 63–77.

T. Hastie, R. Tibshirani, and J. Friedman (2009). The elements of statistical learning: data mining, inference, and prediction. Springer Science & Business Media.

T. Hastie, R. Tibshirani, and M. Wainwright (2015). Statistical Learning with Sparsity: The Lasso and Generalizations. CRC Press.

Q. Huang, T. Xia, H. Sun, M. Yamada, and Y. Chang (2020). Unsupervised Nonlinear Feature Selection from High-Dimensional Signed Networks. *In Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):4182-4189.

Q. Huang, M. Yamada, Y. Tian, D. Singh, D. Yin, and Y. Chang (2020). Graphlime: Local interpretable model explanations for graph neural networks. *arXiv preprint arXiv*:2001.06216.

J. Palmer, K. Kreutz-Delgado, B. D. Rao, and D. P. Wipf (2006). Variational EM algorithms for non-Gaussian latent variable models. *In Advances in neural information processing systems*, 1059–1066.

T. Park and G. Casella (2008). The bayesian lasso. *Journal of the American Statistical Association*, 103(482):681–686.

H. Peng, F. Long, and C. Ding (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, 27(8):1226–1238.

B. Poignard and M. Yamada (2020). Sparse Hilbert-Schmidt Independence Criterion Regression. *In International Conference on Artificial Intelligence and Statistics*, 538–548.

N. Shervashidze and F. Bach (2015). Learning the structure for structured sparsity. *IEEE Transactions on Signal Processing*, 63(18):4894-4902.

D. Singh, H. Climente-González, M. Petrovich, E. Kawakami, and M. Yamada (2020). Fsnet: Feature selection network on high-dimensional biological data. *arXiv preprint arXiv*:2001.08322.

L. Song, A. Smola, A. Gretton, J. Bedo, and K. Borgwardt (2012). Feature selection via dependence maximization. *The Journal of Machine Learning Research*, 13(1):1393–1434.

G. J. Székely and M. L. Rizzo (2009). Brownian distance covariance. *The annals of applied statistics*, 1236–1265.

R. Tibshirani (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society (Series B)*, 58:267–288.

R. Tomioka, T. Suzuki, and M. Sugiyama (2011). Super-Linear Convergence of Dual Augmented Lagrangian Algorithm for Sparsity Regularized Estimation. *Journal of Machine Learning Research*, 12(5).

S. V. Erp, D. L. Oberski, and J. Mulder (2019). Shrinkage priors for bayesian penalized regression. *Journal of Mathematical Psychology*, 89:31–50.

J. Walters-Williams and Y. Li (2009). Estimation of mutual information: A survey. *In International Conference on Rough Sets and Knowledge Technology*, 389–396.

M. Yamada, W. Jitkrittum, L. Sigal, E. P. Xing, and M. Sugiyama (2014). High-dimensional feature selection by feature-wise kernelized lasso. *Neural computation*, 26(1):185–207.

M. Yamada, J. Tang, J. Lugo-Martinez, E. Hodzic, R. Shrestha, A. Saha, H. Ouyang, D. Yin, H. Mamitsuka, C. Sahinalp, P. Radivojac, F. Menczer, and Y. Chang (2018). Ultra high-dimensional nonlinear feature selection for big biological data. *IEEE Transactions on Knowledge and Data Engineering*, 30(7):1352–1365.

M. Yamada, Y. Umezu, K. Fukumizu, and I. Takeuchi (2018). Post selection inference with kernels. *In International Conference on Artificial Intelligence and Statistics*, 152-160, PMLR.

M. Yamada, D. Wu, Y. H. H. Tsai, H. Ohta, R. Salakhutdinov, I. Takeuchi, and K. Fukumizu (2018). Post Selection Inference with Incomplete Maximum Mean Discrepancy Estimator. *In International Conference on Learning Representations.*

Q. Zhang, S. Filippi, A. Gretton, and D. Sejdinovic (2018). Large-scale kernel methods for independence testing. *Statistics and Computing*, 28(1):113–130.

Z. Zhao, L. Wang, and H. Liu (2010). Efficient spectral feature selection with minimum redundancy. *In Proceedings of the AAAI Conference on Artificial Intelligence*, 24(1).

H. Zou and T. Hastie (2005). Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2):301–320.

H. Zou (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429.

Supplementary Material: Effective Nonlinear Feature Selection Method based on HSIC Lasso and with Variational Inference

A DERIVATION FOR VARIATIONAL BOUND

This section supplements Sections 3.3 and 3.4.

Equation (8): variational representation for $p(\omega_p|\eta_p)$.

$$\begin{aligned} p(\omega_p|\eta_p) &= q(|\omega_p|\eta_p^{\frac{1}{2}})\eta_p^{\frac{1}{2}} \\ &= \eta_p^{\frac{1}{2}} \exp(\log q(|\omega_p|\eta_p^{\frac{1}{2}})) \\ &= \eta_p^{\frac{1}{2}} \sup_{s_p>0} \exp\left(-\frac{(\omega_p)^2\eta_p}{2s_p} - \phi(s_p)\right) \\ &= \eta_p^{\frac{1}{2}} \sup_{s_p>0} \left\{\mathcal{N}\left(\omega_p \middle| 0, \frac{s_p^t}{\eta_p}\right)\left(\frac{2\pi s_p}{\eta_p}\right)^{\frac{1}{2}} \exp(-\phi(s_p))\right\} \end{aligned}$$

Equation (9): variational bound for $\log p(\bar{\boldsymbol{\kappa}}_{\boldsymbol{y}}|\boldsymbol{\eta})$.

$$\begin{aligned} \log p(\bar{\boldsymbol{\kappa}}_{\boldsymbol{y}}|\boldsymbol{\eta}) &= \log \int_{\mathbb{R}^P} \mathcal{N}\left(\bar{\boldsymbol{\kappa}}_{\boldsymbol{y}} \middle| \bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}\boldsymbol{\omega}, \sigma^2\boldsymbol{I}\right) p(\boldsymbol{\omega}|\boldsymbol{\eta}) d\boldsymbol{\omega} \\ &= \log \int_{\mathbb{R}^P} \mathcal{N}\left(\bar{\boldsymbol{\kappa}}_{\boldsymbol{y}} \middle| \bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}\boldsymbol{\omega}, \sigma^2\boldsymbol{I}\right) \prod_{p=1}^{P} \eta_p^{\frac{1}{2}} \sup_{s_p>0} \left\{\mathcal{N}\left(\omega_p \middle| 0, \frac{s_p}{\eta_p}\right)\left(\frac{2\pi s_p}{\eta_p}\right)^{\frac{1}{2}} \exp(-\phi(s_p))\right\} d\boldsymbol{\omega} \\ &= \log \int_{\mathbb{R}^P} \sup_{\boldsymbol{s}\in\mathbb{R}_+^P} \mathcal{N}\left(\bar{\boldsymbol{\kappa}}_{\boldsymbol{y}} \middle| \bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}\boldsymbol{\omega}, \sigma^2\boldsymbol{I}\right) \mathcal{N}\left(\boldsymbol{\omega} \middle| \boldsymbol{0}, \boldsymbol{\Xi}^{-1}\right) \prod_{p=1}^{P} \eta_p^{\frac{1}{2}} \left(\frac{2\pi s_p}{\eta_p}\right)^{\frac{1}{2}} \exp(-\phi(s_p)) d\boldsymbol{\omega} \\ &\geq \sup_{\boldsymbol{s}\in\mathbb{R}_+^P} \left[\log\left\{\int_{\mathbb{R}^P} \mathcal{N}\left(\bar{\boldsymbol{\kappa}}_{\boldsymbol{y}} \middle| \bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}\boldsymbol{\omega}, \sigma^2\boldsymbol{I}\right) \mathcal{N}\left(\boldsymbol{\omega} \middle| \boldsymbol{0}, \boldsymbol{\Xi}^{-1}\right) d\boldsymbol{\omega}\right\} + \log \prod_{p=1}^{P} \eta_p^{\frac{1}{2}} \left(\frac{2\pi s_p}{\eta_p}\right)^{\frac{1}{2}} \exp(-\phi(s_p))\right] \\ &= \sup_{\boldsymbol{s}\in\mathbb{R}_+^P} \left[\log \mathcal{N}\left(\bar{\boldsymbol{\kappa}}_{\boldsymbol{y}} \middle| \boldsymbol{0}, \bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}\boldsymbol{\Xi}^{-1}\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}^\top + \sigma^2\boldsymbol{I}\right) + \sum_{p=1}^{P}\left\{\frac{1}{2}\log\eta_p + \frac{1}{2}\log(2\pi) + \frac{1}{2}\log\left(\frac{s_p}{\eta_p}\right) - \phi(s_p)\right\}\right] \\ &= -\inf_{\boldsymbol{s}\in\mathbb{R}_+^P} \left[\log \frac{\exp\left\{-\frac{1}{2}\bar{\boldsymbol{\kappa}}_{\boldsymbol{y}}^\top \left(\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}\boldsymbol{\Xi}^{-1}\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}^\top + \sigma^2\boldsymbol{I}\right)^{-1} \bar{\boldsymbol{\kappa}}_{\boldsymbol{y}}\right\}}{\left\{(2\pi)^{N^2} \det\left(\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}\boldsymbol{\Xi}^{-1}\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}^\top + \sigma^2\boldsymbol{I}\right)\right\}^{\frac{1}{2}}} - \frac{1}{2}\log\det\boldsymbol{\Xi} \right. \\ &\qquad \left. + \sum_{p=1}^{P}\left\{\frac{1}{2}\log\eta_p - \phi(s_p)\right\} + \frac{P}{2}\log(2\pi)\right] \\ &= -\inf_{\boldsymbol{s}\in\mathbb{R}_+^P} \left[-\frac{1}{2}\bar{\boldsymbol{\kappa}}_{\boldsymbol{y}}^\top \left(\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}\boldsymbol{\Xi}^{-1}\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}^\top + \sigma^2\boldsymbol{I}\right)^{-1} \bar{\boldsymbol{\kappa}}_{\boldsymbol{y}} - \frac{1}{2}\log\det\left(\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}\boldsymbol{\Xi}^{-1}\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}^\top + \sigma^2\boldsymbol{I}\right) - \frac{1}{2}\log\det\boldsymbol{\Xi} \right. \\ &\qquad \left. + \sum_{p=1}^{P}\left\{\frac{1}{2}\log\eta_p - \phi(s_p)\right\} + \frac{P-N^2}{2}\log(2\pi)\right] \end{aligned}$$

where $\boldsymbol{\Xi} \in \mathbb{R}^{P\times P}$ is a diagonal matrix such that $[\boldsymbol{\Xi}]_{pp} = \eta_p/s_p$, and the inequality is based on the interchange of integral and supremum.

The first term on the right side is the solution to the following optimization problem.

$$\frac{1}{2}\bar{\boldsymbol{\kappa}}_{\boldsymbol{y}}^{\top}\left(\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}\boldsymbol{\Xi}^{-1}\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}^{\top}+\sigma^{2}\boldsymbol{I}\right)^{-1}\bar{\boldsymbol{\kappa}}_{\boldsymbol{y}}=\inf_{\boldsymbol{\mu}\in\mathbb{R}^{P}}\left\{\frac{1}{2\sigma^{2}}\left\|\bar{\boldsymbol{\kappa}}_{\boldsymbol{y}}-\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}\boldsymbol{\mu}\right\|_{2}^{2}+\frac{1}{2}\boldsymbol{\mu}^{\top}\boldsymbol{\Xi}\boldsymbol{\mu}\right\}$$

The second and third terms on the right side are the solution to the following optimization problem. We use the properties of the Schur complementary.

$$\begin{aligned}&\frac{1}{2}\log\det\left(\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}\boldsymbol{\Xi}^{-1}\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}^{\top}+\sigma^{2}\boldsymbol{I}\right)+\frac{1}{2}\log\det\boldsymbol{\Xi}\\&=\frac{1}{2}\log\det\left(\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}\boldsymbol{\Xi}^{-1}\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}^{\top}+\sigma^{2}\boldsymbol{I}\right)\det\boldsymbol{\Xi}\\&=\frac{1}{2}\log\det\left(\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}^{\top}\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}+\sigma^{2}\boldsymbol{\Xi}\right)+\frac{N^{2}-P}{2}\log(\sigma^{2})\\&=\inf_{\boldsymbol{\Sigma}\succeq0}\frac{1}{2}\mathrm{Tr}\left\{\left(\frac{1}{\sigma^{2}}\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}^{\top}\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}+\boldsymbol{\Xi}\right)\boldsymbol{\Sigma}\right\}-\frac{1}{2}\log\det\left(\frac{1}{\sigma^{2}}\boldsymbol{\Sigma}\right)+\frac{N^{2}-P}{2}\log(\sigma^{2})-\frac{P}{2}\\&=\inf_{\boldsymbol{\Sigma}\succeq0}\frac{1}{2\sigma^{2}}\mathrm{Tr}\left(\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}^{\top}\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}\boldsymbol{\Sigma}\right)+\frac{1}{2}\mathrm{Tr}\left(\boldsymbol{\Xi}\boldsymbol{\Sigma}\right)-\frac{1}{2}\log\det\boldsymbol{\Sigma}+\frac{N^{2}}{2}\log(\sigma^{2})-\frac{P}{2}\end{aligned}$$

Using these equations, (9) is derived.

B DERIVATION FOR UPDATE EQUATIONS

This section supplements Section 3.6.

Update equations (12), (13), (14), (15), and (16).

Adding the optimization problem for $\boldsymbol{\eta}$ and σ^2 from (11), the variational bound is rewritten as

$$\inf_{\boldsymbol{\mu}\in\mathbb{R}_{+}^{P}}\inf_{\boldsymbol{\Sigma}\succeq0}\inf_{\boldsymbol{s}\in\mathbb{R}_{+}^{P}}\inf_{\boldsymbol{\eta}\in\mathbb{R}_{+}^{P}}\inf_{\sigma^{2}>0}\left\{f(\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{s},\boldsymbol{\eta},\sigma^{2})+\lambda\|\boldsymbol{\mu}\|_{1}\right\}.$$

We then extract the local minimization problems. Equation (12) is trivial and becomes an optimization problem similar to Elastic-Net. Equation (13) is derived as follows.

$$\frac{\partial}{\partial\boldsymbol{\Sigma}}\left\{\frac{1}{2\sigma^{2}}\mathrm{Tr}\left(\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}^{\top}\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}\boldsymbol{\Sigma}\right)+\frac{1}{2}\mathrm{Tr}\left(\boldsymbol{\Xi}\boldsymbol{\Sigma}\right)-\frac{1}{2}\log\det\boldsymbol{\Sigma}\right\}=\frac{1}{2}\left\{\frac{1}{\sigma^{2}}\left(\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}^{\top}\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}\right)+\boldsymbol{\Xi}-\boldsymbol{\Sigma}^{-1}\right\}$$

Therefore, the local optimal solution is

$$\boldsymbol{\Sigma}=\sigma^{2}\left(\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}^{\top}\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}+\sigma^{2}\boldsymbol{\Xi}\right)^{-1}.$$

Equation (14) is also trivial, but the local optimal solution depends on ϕ derived from the prior distribution on the basis of (7). Equation (15) is derived as follows.

$$\frac{\partial}{\partial\boldsymbol{\eta}}\left\{\frac{1}{2}\boldsymbol{\mu}^{\top}\boldsymbol{\Xi}\boldsymbol{\mu}+\frac{1}{2}\mathrm{Tr}\left(\boldsymbol{\Xi}\boldsymbol{\Sigma}\right)-\frac{1}{2}\sum_{p=1}^{P}\log\eta_{p}\right\}=\frac{1}{2}\left\{\boldsymbol{\mu}\odot\boldsymbol{\mu}\oslash\boldsymbol{s}+\mathrm{diag}\boldsymbol{\Sigma}\oslash\boldsymbol{s}-\frac{1}{\boldsymbol{\eta}}\right\}$$

Therefore, the local optimal solution is

$$\boldsymbol{\eta}=\boldsymbol{s}\oslash(\boldsymbol{\mu}\odot\boldsymbol{\mu}+\mathrm{diag}\boldsymbol{\Sigma}).$$

Equation (16) is derived as follows.

$$\frac{\partial}{\partial\sigma^{2}}\left\{\frac{1}{2\sigma^{2}}\left\|\bar{\boldsymbol{\kappa}}_{\boldsymbol{y}}-\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}\boldsymbol{\mu}\right\|_{2}^{2}+\frac{1}{2\sigma^{2}}\mathrm{Tr}\left(\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}^{\top}\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}\boldsymbol{\Sigma}\right)+\frac{N^{2}}{2}\log(\sigma^{2})\right\}=-\frac{1}{2(\sigma^{2})^{2}}\left\{\left\|\bar{\boldsymbol{\kappa}}_{\boldsymbol{y}}-\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}\boldsymbol{\mu}\right\|_{2}^{2}+\mathrm{Tr}\left(\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}^{\top}\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}\boldsymbol{\Sigma}\right)-N^{2}\sigma^{2}\right\}$$

Therefore, the local optimal solution is

$$\sigma^{2}=\frac{1}{N^{2}}\left(\left\|\bar{\boldsymbol{\kappa}}_{\boldsymbol{y}}-\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}\boldsymbol{\mu}\right\|_{2}^{2}+\mathrm{Tr}\left(\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}^{\top}\bar{\boldsymbol{\kappa}}_{\boldsymbol{X}}\boldsymbol{\Sigma}\right)\right).$$

C DETAILS FOR PRIOR DISTRIBUTIONS

This section supplements Section 3.7 and 3.8.

C.1 Variational Representations

Defining $g(u) := -\log q(u^{\frac{1}{2}})$, $\phi(s)$ satisfying (7), can be represented as follows:

$$\phi(s) = -g^*\left(\frac{1}{2s}\right)$$

where g^* is the concave conjugate of g.

The $-\log q(u) = g(u^2)$ can be represented in the following convex variational form from Theorem 1 of Palmer et al. (2006),

$$\begin{aligned}-\log q(u) &= -\sup_{\xi>0} \log \mathcal{N}\left(u \,\middle|\, 0, \xi^{-1}\right) \varphi(\xi)\\ &= -\sup_{\xi>0}\left\{-\frac{u^2\xi}{2} + \log \xi^{\frac{1}{2}}\varphi(\xi) - \frac{1}{2}\log 2\pi\right\}\end{aligned}$$

if and only if $g(u) = -\log q(u^{\frac{1}{2}})$ is concave on $\mathbb{R}_+$. In this case,

$$\varphi(\xi) = \sqrt{\frac{2\pi}{\xi}} \exp\left(g^*\left(\frac{\xi}{2}\right)\right).$$

Thus, as $\xi = s^{-1}$,

$$\log q(u) = \sup_{s>0}\left\{-\frac{u^2}{2s} + \log\left(\frac{1}{s}\right)^{\frac{1}{2}} \varphi\left(\frac{1}{s}\right) - \frac{1}{2}\log 2\pi\right\}.$$

Consequently, $\phi(s)$ can be rewritten from (7) as

$$\begin{aligned}\phi(s) &= -\log\left(\frac{1}{s}\right)^{\frac{1}{2}} \varphi\left(\frac{1}{s}\right) + \frac{1}{2}\log 2\pi\\ &= -\log\left(\frac{1}{s}\right)^{\frac{1}{2}} \sqrt{2\pi s}\exp\left(g^*\left(\frac{1}{2s}\right)\right) + \frac{1}{2}\log 2\pi\\ &= -g^*\left(\frac{1}{2s}\right)\end{aligned}$$

C.2 Prior: Generalized Gaussian

The generalized Gaussian, such as

$$p(\omega_p|\eta_p, \alpha, \beta) = \frac{\eta_p^{\frac{1}{2}}\beta}{2\alpha\Gamma\left(\frac{1}{\beta}\right)} \exp\left(-\left(\frac{|\omega_p|\eta_p^{\frac{1}{2}}}{\alpha}\right)^\beta\right)$$

where $\alpha \in \mathbb{R}_+$ and $\beta \in \mathbb{R}_+$ are the scale and shape parameters respectively, can be also rewritten in this variational representation for $0 < \beta \leq 2$. In this case, $q(u)$ can be rewritten as

$$q(u) = \frac{\beta}{2\alpha\Gamma\left(\frac{1}{\beta}\right)} \exp\left(-\left(\frac{|u|}{\alpha}\right)^\beta\right)$$

Therefore, $g(u)$ can be obtained as

$$g(u) = \left(\frac{u^{\frac{1}{2}}}{\alpha}\right)^\beta - \log\beta + \log 2\alpha\Gamma\left(\frac{1}{\beta}\right)$$

and its concave conjugate g^* can be represented as

$$g^*(p) = \left\{ \left(\frac{2\alpha^\beta}{\beta}\right)^{\frac{2}{\beta-2}} - \frac{1}{\alpha^\beta}\left(\frac{2\alpha^\beta}{\beta}\right)^{\frac{\beta}{\beta-2}} \right\} p^{\frac{\beta}{\beta-2}} + \log\beta - \log 2\alpha\Gamma\left(\frac{1}{\beta}\right)$$

where p is a conjugate variable of u, satisfying $p = g'(u)$. Consequently, $\phi(s_p)$ can be rewritten as

$$\phi(s_p) = \left\{ \frac{1}{\alpha^\beta}\left(\frac{2\alpha^\beta}{\beta}\right)^{\frac{\beta}{\beta-2}} - \left(\frac{2\alpha^\beta}{\beta}\right)^{\frac{2}{\beta-2}} \right\} (2s_p)^{-\frac{\beta}{\beta-2}} - \log\beta + \log 2\alpha\Gamma\left(\frac{1}{\beta}\right)$$

In particular, the Laplace distribution, which is often used in Bayesian Lasso (Park and Casella, 2008), corresponds to $\beta = 1$, and $\phi(s_p)$ can be simply rewritten as

$$\phi(s_p) = \frac{s_p}{2\alpha^2} + \log 2\alpha$$

Additionally, (14) can be rewritten in the closed-form update equation (19), which is derived as follows.

$$\frac{\partial}{\partial \boldsymbol{s}}\left\{\frac{1}{2}\boldsymbol{\mu}^\top\boldsymbol{\Xi}\boldsymbol{\mu} + \frac{1}{2}\mathrm{Tr}(\boldsymbol{\Xi}\boldsymbol{\Sigma}) + \frac{\boldsymbol{s}}{2\alpha^2}\right\} = \frac{1}{2}\left\{\frac{\boldsymbol{s}\odot\boldsymbol{s}}{\alpha^2} - \boldsymbol{\eta}\odot(\boldsymbol{\mu}\odot\boldsymbol{\mu} + \mathrm{diag}\boldsymbol{\Sigma})\right\} \oslash (\boldsymbol{s}\odot\boldsymbol{s})$$

Therefore, the local optimal solution is

$$\boldsymbol{s} = \alpha\sqrt{\boldsymbol{\eta}\odot(\boldsymbol{\mu}\odot\boldsymbol{\mu} + \mathrm{diag}\boldsymbol{\Sigma})}$$

C.3 Prior: Student's t

Student's t can be rewritten in this variational representation. We consider the following density,

$$p(\omega_p|\eta_p, \nu) = \left(\frac{\eta_p}{2\pi}\right)^{\frac{1}{2}} \frac{\Gamma\left(\nu + \frac{1}{2}\right)}{\Gamma\left(\nu\right)} \left(1 + \frac{\eta_p\omega_p^2}{2}\right)^{-\nu-\frac{1}{2}},$$

and $q(u)$ can be rewritten as

$$q(u) = \left(\frac{1}{2\pi}\right)^{\frac{1}{2}} \frac{\Gamma\left(\nu + \frac{1}{2}\right)}{\Gamma\left(\nu\right)} \left(1 + \frac{u^2}{2}\right)^{-\nu-\frac{1}{2}}.$$

Therefore, $g(u)$ can be obtained as

$$g(u) = \left(\nu + \frac{1}{2}\right)\log\left(1 + \frac{u}{2}\right) - \log\left(\frac{\Gamma\left(\nu+\frac{1}{2}\right)}{\Gamma\left(\nu\right)}\right) + \frac{1}{2}\log\left(2\pi\right),$$

and its concave conjugate g^* can be represented as

$$g^*(p) = -2p + \left(\nu + \frac{1}{2}\right)\log\left(2p\right) + \log\left(\frac{\Gamma\left(\nu+\frac{1}{2}\right)}{\Gamma\left(\nu\right)}\right) - \left(\nu + \frac{1}{2}\right)\left(\log\left(\nu + \frac{1}{2}\right) - 1\right) - \frac{1}{2}\log\left(2\pi\right)$$

where p is a conjugate variable of u, satisfying $p = g'(u)$. Consequently, $\phi(s_p)$ can be rewritten as

$$\phi(s_p) = \frac{1}{s_p} + \left(\nu + \frac{1}{2}\right)\log\left(s_p\right) - \log\left(\frac{\Gamma\left(\nu+\frac{1}{2}\right)}{\Gamma(\nu)}\right) + \left(\nu + \frac{1}{2}\right)\left(\log\left(\nu + \frac{1}{2}\right) - 1\right) + \frac{1}{2}\log(2\pi)$$

Additionally, (14) can be rewritten in the closed-form update equation (23), which is derived as follows.

$$\frac{\partial}{\partial \boldsymbol{s}}\left\{\frac{1}{2}\boldsymbol{\mu}^\top\boldsymbol{\Xi}\boldsymbol{\mu} + \frac{1}{2}\mathrm{Tr}(\boldsymbol{\Xi}\boldsymbol{\Sigma}) + \frac{1}{\boldsymbol{s}} + \left(\nu + \frac{1}{2}\right)\log\left(\boldsymbol{s}\right)\right\} = \left\{\left(\nu + \frac{1}{2}\right)\boldsymbol{s} - \left(1 + \frac{\boldsymbol{\eta}}{2}\odot(\boldsymbol{\mu}\odot\boldsymbol{\mu} + \mathrm{diag}\boldsymbol{\Sigma})\right)\right\} \oslash (\boldsymbol{s}\odot\boldsymbol{s})$$

Therefore, the local optimal solution is

$$\boldsymbol{s} = \frac{1}{\nu + \frac{1}{2}}\left\{1 + \frac{\boldsymbol{\eta}}{2}\odot(\boldsymbol{\mu}\odot\boldsymbol{\mu} + \mathrm{diag}\boldsymbol{\Sigma})\right\}$$

D ESTIMATION OF COMPUTATIONAL COMPLEXITY

This section supplements Section 3.9. We estimate the computational complexity of update equations.

- Equation (12) for $\boldsymbol{\omega}$: $\mathcal{O}(MN^3P)$
  (if we use LARS and select M features)
- Equation (13) for $\boldsymbol{\Sigma}$: $\mathcal{O}(P^3)$
  (but we can compute relatively fast with the Cholesky decomposition)
- Equation (14) for $\boldsymbol{s}$: $\mathcal{O}(P)$
  (if we use Student's t for prior)
- Equation (15) for $\boldsymbol{\eta}$: $\mathcal{O}(P)$
- Equation (16) for σ^2: $\mathcal{O}(N^2P + P^2)$

Therefore, we can estimate the computational complexity per one iteration as $\mathcal{O}(MN^3P + P^3)$.