Source Separation with Gaussian Process Models

Sunho Park and Seungjin Choi

Department of Computer Science
Pohang University of Science and Technology
San 31 Hyoja-dong, Nam-gu, Pohang 790-784, Korea
{titan,seungjin}@postech.ac.kr

Abstract. In this paper we address a method of source separation in the case where sources have certain temporal structures. The key contribution in this paper is to incorporate Gaussian process (GP) model into source separation, representing the latent function which characterizes the temporal structure of a source by a random process with Gaussian prior. Marginalizing out the latent function leads to the Gaussian marginal likelihood of source that is plugged in the mutual information-based loss function for source separation. In addition, we also consider the leave-one-out predictive distribution of source, instead of the marginal likelihood, in the same framework. Gradient-based optimization is applied to estimate the demixing matrix through the mutual information minimization, where the marginal distribution of source is replaced by the marginal likelihood of the source or its leave-one-out predictive distribution. Numerical experiments confirm the useful behavior of our method, compared to existing source separation methods.

1 Introduction

Source separation assumes that multivariate observation data $\boldsymbol{x}_t = [x_{1,t} \cdots x_{n,t}]^\top$ ($x_{i,t}$ represents the ith element of $\boldsymbol{x}_t \in \mathbb{R}^n$) are generated by

$$\boldsymbol{x}_t = \boldsymbol{A}\boldsymbol{s}_t, \tag{1}$$

where $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is the mixing matrix and $\boldsymbol{s}_t = [s_{1,t}, \ldots, s_{n,t}]^\top$ is the source vector whose elements are assumed to statistically independent. Source separation is an unsupervised learning task, the goal of which is to restore unknown independent sources $\boldsymbol{s}_t$ up to scaling and permutation ambiguities, without the knowledge of the invertible mixing matrix $\boldsymbol{A}$, given a set of data points, $\{\boldsymbol{x}_t\}_{t=1}^N$. In other words, source separation aims to estimate a demixing matrix $\boldsymbol{W}$ such that $\boldsymbol{W}\boldsymbol{A} = \boldsymbol{P}\boldsymbol{\Lambda}$ is a *transparent transform*, where $\boldsymbol{P}$ is the permutation matrix and $\boldsymbol{\Lambda}$ is an arbitrary invertible diagonal matrix.

Various methods for source separation have been developed (for example, see [1] and references therein). Two exemplary independent component analysis (ICA) methods might be Infomax [2] and FastICA [3] where only spatial independence is exploited, assuming that sources follow non-Gaussian distributions.

Infomax is indeed maximum likelihood source separation where sources are latent variables that are treated as nuisance parameters [4]. In cases where individual source is temporally correlated, it is well known that second-order statistics (e.g., time-delayed correlations) is sufficient to achieve separation. SOBI [5] is a widely-used algebraic method where a set of several time-delayed correlation matrices of whitened data is jointly diagonalized by a unitary transform in order to estimate a demixing matrix. Alternatively, a linear latent function of parametric form (e.g., auto-regressive (AR) model) was often used as a source generative model in order to characterize the temporal structure of sources [6–8]. In such cases, parameters involving AR source generative models should be estimated in learning a mixing matrix or a demixing matrix.

Gaussian process (GP) model is a nonparametric method, which recently attracts extensive interests in machine learning. For a recent tutorial, see [9, 10] with references therein. In this paper we use a GP model to characterize the temporal structure of a source, representing the latent function (which relates the current sample of source to past samples) by a random process with Gaussian prior. The marginal likelihood of source is Gaussian, which is computed by integrating out the latent function. We incorporate the GP source model into source separation based on the mutual information minimization, modeling the probability distribution of source by the marginal likelihood of source in the mutual information-based loss function. Alternatively, we also consider the leave-one-out (LOO) predictive distribution, instead of the marginal likelihood of source, in the same framework. We use a gradient-based optimization to estimate the demixing matrix, through the mutual information minimization, where the marginal likelihood of source or LOO predictive distribution of source is used to model the marginal entropy of source.

2 GP Models for Sources

The latent function $f_i(\cdot)$ relates the current sample of source $s_{i,t}$ to past p samples, leading to

$$s_{i,t} = f_i\left(\boldsymbol{s}_{i,t-1:t-p}^\top\right) + \varepsilon_{i,t}, \tag{2}$$

where $\boldsymbol{s}_{i,t-1:t-p} = [s_{i,t-1}, s_{i,t-2}, \ldots, s_{i,t-p}]$ is a collection of past p samples and $\epsilon_{i,t}$ is the white Gaussian noise with zero mean and unit variance, i.e., $\epsilon_{i,t} \sim \mathcal{G}(\epsilon_{i,t}; 0, 1)$. In the case of linear AR model, the latent function is parameterized by

$$f_i\left(\boldsymbol{s}_{i,t-1:t-p}^\top\right) = \sum_{\tau=1}^{p} h_{i,\tau} s_{i,t-\tau}, \tag{3}$$

where $h_{i,\tau}$ are AR coefficients.

GP model represents the latent function $f_i(\cdot)$ by a random process with Gaussian prior, unlike AR model employs the parametric form (3). We place a

GP prior over the function $f_i(\cdot)$, i.e.,

$$f_i \sim \mathcal{GP}\left(0, k\left(\boldsymbol{s}_{i,t:t-p+1}^\top, \boldsymbol{s}_{i,\tau:\tau-p+1}^\top\right)\right), \tag{4}$$

where $k\left(\boldsymbol{s}_{i,t:t-p+1}^\top, \boldsymbol{s}_{i,\tau:\tau-p+1}^\top\right)$ is a *covariance function*. We use the squared exponential covariance function, i.e.,

$$k\left(\boldsymbol{s}_{i,t:t-p+1}^\top, \boldsymbol{s}_{i,\tau:\tau-p+1}^\top\right) = \exp\left\{-\lambda_i \|\boldsymbol{s}_{i,t:t-p+1}^\top - \boldsymbol{s}_{i,\tau:\tau-p+1}^\top\|^2\right\}, \tag{5}$$

where λ_i is a length-scale hyperparameter.

We refer to the source generative model (2) with GP prior (4) as *GP source model*. The GP source model follows the standard GP regression in which $\boldsymbol{s}_{i,1:N}^\top = [s_{i,1}, \ldots, s_{i,N}]^\top$ is a collection of responses and $\mathcal{S}_i = \left\{\boldsymbol{s}_{i,t-1:t-p}^\top\right\}_{t=1}^N$ is a set of regressors.

We define the vector $\boldsymbol{f}_i \in \mathbb{R}^N$ as

$$\boldsymbol{f}_i = [f_{i,0}, f_{i,1}, \ldots, f_{i,N-1}]^\top,$$

where $f_{i,t} = f_i\left(\boldsymbol{s}_{i,t:t-p+1}^\top\right)$. Then the likelihood of source i is given by

$$p\left(\boldsymbol{s}_{i,1:N}^\top | \boldsymbol{f}_i, \mathcal{S}_i\right) = \mathcal{G}(\boldsymbol{s}_{i,1:N}^\top; \boldsymbol{f}_i, \boldsymbol{I}_N), \tag{6}$$

where $\boldsymbol{I}_N$ is the $N \times N$ identity matrix. Then the marginal likelihood of source i is obtained by integrating the likelihood times the prior

$$p_i\left(\boldsymbol{s}_{i,1:N}^\top | \mathcal{S}_i\right) = \int p\left(\boldsymbol{s}_{i,1:N}^\top | \boldsymbol{f}_i, \mathcal{S}_i\right) p\left(\boldsymbol{f}_i | \mathcal{S}_i\right) d\boldsymbol{f}_i, \tag{7}$$

where the prior is given by (4),

$$p\left(\boldsymbol{f}_i | \mathcal{S}_i\right) = \mathcal{G}(\boldsymbol{f}_i; 0, \boldsymbol{K}_i),$$

where $\boldsymbol{K}_i$ is a $N \times N$ matrix whose (u, v)-element is given by

$$[\boldsymbol{K}_i]_{u,v} = k(\boldsymbol{s}_{i,u-1:u-p}, \boldsymbol{s}_{i,v-1:v-p}).$$

The log of the marginal likelihood, denoted by $\log p_i^{ML}$, is of the form

$$\begin{aligned}\log p_i^{ML}\left(\boldsymbol{s}_{i,1:N}^\top\right) &= \log p\left(\boldsymbol{s}_{i,1:N}^\top | \mathcal{S}_i\right) \\ &= -\frac{1}{2}\boldsymbol{s}_{i,1:N}\boldsymbol{\Sigma}_i^{-1}\boldsymbol{s}_{i,1:N}^\top - \frac{1}{2}\log|\boldsymbol{\Sigma}_i| - \frac{N}{2}\log 2\pi,\end{aligned} \tag{8}$$

where $\boldsymbol{\Sigma}_i = \boldsymbol{K}_i + \boldsymbol{I}_N$.

We also consider the LOO predictive distribution which is Gaussian:

$$p_i^{LOO}\left(\boldsymbol{s}_{i,1:N}^\top\right) = \prod_{t=1}^N p\left(s_{i,t} | \mathcal{S}_i, \boldsymbol{s}_{i,1:N}^{-t}\right) = \prod_{t=1}^N \mathcal{G}(s_{i,t}; \mu_{i,t}, \sigma_{i,t}^2), \tag{9}$$

where $\boldsymbol{s}_{i,1:N}^{-t} = [s_{i,1}, \ldots, s_{i,t-1}, s_{i,t+1}, \ldots, s_{i,N}]$ denotes all samples of source i but $s_{i,t}$. The LOO predictive mean $\mu_{i,t}$ and variance $\sigma_{i,t}^2$ are given by

$$\mu_{i,t} = s_{i,t} - \left[\boldsymbol{\Sigma}_i^{-1}\boldsymbol{s}_{i,1:N}^\top\right]_t / \left[\boldsymbol{\Sigma}_i^{-1}\right]_{t,t},$$
$$\sigma_{i,t}^2 = 1/\left[\boldsymbol{\Sigma}_i^{-1}\right]_{t,t}.$$

Thus the log of LOO predictive distribution is of the form

$$\log p_i^{LOO}(\boldsymbol{s}_{i,1:N}^\top) = -\frac{1}{2}\sum_{t=1}^{N}\left\{-\log\left[\boldsymbol{\Sigma}_i^{-1}\right]_{t,t} + \frac{\left(\left[\boldsymbol{\Sigma}_i^{-1}\boldsymbol{s}_{i,1:N}^\top\right]_t\right)^2}{\left[\boldsymbol{\Sigma}_i^{-1}\right]_{t,t}} + \log 2\pi\right\} \quad (10)$$

The log LOO predictive distribution (10) is often referred to as *log pseudo-likelihood* that is an approximation of the log marginal likelihood (8)[11]. The marginal likelihood or LOO predictive distribution is used to learn hyperparameters in GP regression. We use the marginal likelihood or the LOO predictive distribution as an estimate of the source distribution which is required in the source separation based on the mutual information minimization. It is known that the LOO predictive distribution is more robust to model mis-specification, compared to [12, 10]. The model mis-specification occurs when the model assumption is not suitable to describe the observation data. In the case of source separation using GP models, the model mis-specification might arise when inappropriate model order p is chosen or the selected kernel function is not suitable. The marginal likelihood represents the probability distribution of source given a certain model assumption, so it might be affected by the model mis-specification. Fortunately the aim in this paper is to estimate a demixing matrix for source separation rather than to estimate hyperparameters for source model fitting. Thus, the mis-specification problem is not critical in source separation. This issue is investigated through experiments (see Sec. 5.3).

3 Source Separation with GP Models

In this section we present the main contribution of this paper, developing methods which incorporate the GP source model (illustrated in Sec 2) into source separation based on the mutual information minimization.

Let us consider the demixing model:

$$\boldsymbol{y}_t = \boldsymbol{W}\boldsymbol{x}_t, \quad (11)$$

where $\boldsymbol{W} \in \mathbb{R}^{n\times n}$ is the demixing matrix. The goal of source separation is to learn the demixing matrix $\boldsymbol{W}$ such that $\boldsymbol{y}_t = \boldsymbol{P}\boldsymbol{\Lambda}\boldsymbol{s}_t$, i.e., $\boldsymbol{W}\boldsymbol{A}$ is a transparent transform.

Sources are assumed to be mutually independent, which satisfies

$$p\left(\boldsymbol{s}_{1,1:N}^\top, ..., \boldsymbol{s}_{n,1:N}^\top\right) = \prod_{i=1}^{n} p_i\left(\boldsymbol{s}_{i,1:N}^\top\right). \quad (12)$$

The factorial model (12) leads to the mutual information-based risk $R(\boldsymbol{W})$ that is of the form

$$\boldsymbol{R}(\boldsymbol{W}) = \mathbb{E}\{L(\boldsymbol{W})\} = \frac{1}{N}\mathbb{E}\left\{\log \frac{p\left(\boldsymbol{y}_{1,1:N}^{\top}, ..., \boldsymbol{y}_{n,1:N}^{\top}\right)}{\prod_{i=1}^{n} p_i\left(\boldsymbol{y}_{i,1:N}^{\top}\right)}\right\}, \tag{13}$$

where $L(\boldsymbol{W})$ is the loss function. The risk (13) is nothing but the normalized Kullback-Leibler divergence between the joint distribution $p\left(\boldsymbol{y}_{1,1:N}^{\top}, ..., \boldsymbol{y}_{n,1:N}^{\top}\right)$ and the product of marginal distributions $\prod_{i=1}^{n} p_i\left(\boldsymbol{y}_{i,1:N}^{\top}\right)$. Since $\boldsymbol{y}_t$ is a linear transform of $\boldsymbol{x}_t$, joint distributions satisfies

$$p\left(\boldsymbol{y}_{1,1:N}^{\top}, ..., \boldsymbol{y}_{n,1:N}^{\top}\right) = \frac{p\left(\boldsymbol{x}_{1,1:N}^{\top}, ..., \boldsymbol{x}_{n,1:N}^{\top}\right)}{|\det \boldsymbol{W}|^N}. \tag{14}$$

Taking this into account, the loss function $L(\boldsymbol{W})$ is given by

$$L(\boldsymbol{W}) = -\log|\det \boldsymbol{W}| - \frac{1}{N}\sum_{i=1}^{n} \log p_i\left(\boldsymbol{y}_{i,1:N}^{\top}\right), \tag{15}$$

where $\frac{1}{N}\log p\left(\boldsymbol{x}_{1,1:N}^{\top}, ..., \boldsymbol{x}_{n,1:N}^{\top}\right)$ is omitted since it does not depend on $\boldsymbol{W}$.

Now we use the log of the marginal likelihood (8) or the log of LOO predictive distribution (10) as an substitute to $\log p_i\left(\boldsymbol{y}_{i,1:N}^{\top}\right)$ in the loss function (15). In the case of the marginal likelihood, the loss function becomes

$$\begin{aligned} L_{ML}(\boldsymbol{W}) &= -\log|\det \boldsymbol{W}| - \frac{1}{N}\sum_{i=1}^{n} \log p_i^{ML}\left(\boldsymbol{y}_{i,1:N}^{\top}\right) \\ &= -\log|\det \boldsymbol{W}| + \frac{1}{2N}\sum_{i=1}^{n}\left\{\boldsymbol{y}_{i,1:N}\boldsymbol{\Sigma}_i^{-1}\boldsymbol{y}_{i,1:N}^{\top} + \log|\boldsymbol{\Sigma}_i|\right\}, \end{aligned} \tag{16}$$

where $\frac{1}{2N}\log 2\pi$ (which does not depend on $\boldsymbol{W}$) is left out. In (16), $\boldsymbol{\Sigma}_i = \boldsymbol{K}_i + \boldsymbol{I}_N$ is computed using $y_{i,t}$ which is the estimate of $s_{i,t}$, i.e.,

$$[\boldsymbol{K}_i]_{u,v} = k\left(\boldsymbol{y}_{i,u-1:u-p}^{\top}, \boldsymbol{y}_{i,v-1:v-p}^{\top}\right).$$

We use a gradient-based optimization to find a solution which minimizes (16). In order to compute the gradient, we first define

$$\boldsymbol{\alpha}_i = \boldsymbol{\Sigma}_i^{-1}\boldsymbol{y}_{i,1:N}^{\top}, \tag{17}$$

$$\boldsymbol{Z}_i^{kl} = \boldsymbol{\Sigma}_i^{-1}\frac{\partial \boldsymbol{K}_i}{\partial w_{k,l}}, \tag{18}$$

where the derivative of the covariance matrix $\boldsymbol{K}_i$ w.r.t $w_{k,l}$ (which is the (k,l)-element of the demixing matrix $\boldsymbol{W}$) is computed as

$$\left[\frac{\partial \boldsymbol{K}_i}{\partial w_{k,l}}\right]_{u,v} = -2\lambda_i k\left(\boldsymbol{y}_{i,u-1:u-p}^{\top}, \boldsymbol{y}_{i,v-1:v-p}^{\top}\right)\left[\boldsymbol{\Delta}\boldsymbol{\Delta}^{\top}\boldsymbol{w}_{i,:}^{\top}\right]_l \delta_{i,k},$$

where $\delta_{i,k}$ is the Kronecker delta, $\boldsymbol{w}_{i,:}$ represents the ith row vector of $\boldsymbol{W}$, and

$$\boldsymbol{\Delta} = [(\boldsymbol{x}_{u-1} - \boldsymbol{x}_{v-1}), \ldots, (\boldsymbol{x}_{u-p} - \boldsymbol{x}_{v-p})]\,.$$

With this definition, the gradient of (16) w.r.t $w_{k,l}$ is determined by

$$\begin{aligned}\frac{\partial L_{ML}}{\partial w_{k,l}} &= -\mathrm{tr}\left\{\boldsymbol{W}^{-1}\frac{\partial \boldsymbol{W}}{\partial w_{k,l}}\right\} + \frac{1}{2N}\sum_{i=1}^{n}\mathrm{tr}\left\{\boldsymbol{y}_{i,1:N}\frac{\partial \boldsymbol{\Sigma}_i^{-1}}{\partial w_{k,l}}\boldsymbol{y}_{i,1:N}^{\top}\right. \\ &\quad \left. + 2\frac{\partial \boldsymbol{y}_{i,1:N}}{\partial w_{k,l}}\boldsymbol{\Sigma}_i^{-1}\boldsymbol{y}_{i,1:N}^{\top} + \boldsymbol{\Sigma}_i^{-1}\frac{\partial \boldsymbol{K}_i}{\partial w_{k,l}}\right\} \\ &= -\mathrm{tr}\left\{\boldsymbol{W}^{-1}\frac{\partial \boldsymbol{W}}{\partial w_{k,l}}\right\} \\ &\quad -\frac{1}{2N}\sum_{i=1}^{n}\delta_{i,k}\mathrm{tr}\left\{\boldsymbol{\alpha}_i\boldsymbol{\alpha}_i^{\top}\frac{\partial \boldsymbol{K}_i}{\partial w_{k,l}} - 2\boldsymbol{x}_{l,1:N}\boldsymbol{\alpha}_i - \boldsymbol{Z}_i^{kl}\right\}. \end{aligned} \tag{19}$$

The hyperparameters λ_i can also be learned through minimizing the loss function and the gradient w.r.t them can be easily computed. However, here we fix them as constant values and learn only demixing matrix. Empirical results show that the performance does not much depend on the values of hyperparameters (see Fig. 1 (a)).

In a similar manner, we also consider the log of LOO predictive distribution (10), leading to the following loss function

$$\begin{aligned}&L_{LOO}(\boldsymbol{W}) \\ &= -\log|\det\boldsymbol{W}| - \frac{1}{N}\sum_{i=1}^{n}\log p_i^{LOO}\left(\boldsymbol{y}_{i,1:N}^{\top}\right) \\ &= -\log|\det\boldsymbol{W}| + \frac{1}{2N}\sum_{i=1}^{n}\sum_{t=1}^{N}\left\{-\log\left[\boldsymbol{\Sigma}_i^{-1}\right]_{t,t} + \frac{\left(\left[\boldsymbol{\Sigma}_i^{-1}\boldsymbol{y}_{i,1:N}^{\top}\right]_t\right)^2}{\left[\boldsymbol{\Sigma}_i^{-1}\right]_{t,t}}\right\}. \end{aligned} \tag{20}$$

The gradient of (20) w.r.t $w_{k,l}$ is calculated by

$$\begin{aligned}\frac{\partial \mathcal{L}_{LOO}}{\partial w_{k,l}} &= -\mathrm{tr}\left\{\boldsymbol{W}^{-1}\frac{\partial \boldsymbol{W}}{\partial w_{k,l}}\right\} \\ &\quad +\frac{1}{2N}\sum_{i=1}^{n}\sum_{t=1}^{N}\frac{\delta_{i,k}}{\left[\boldsymbol{\Sigma}_i^{-1}\right]_{t,t}}\left\{[\boldsymbol{\alpha}_i]_t\left(2\left[\boldsymbol{\Sigma}_i^{-1}\boldsymbol{x}_{l,1:N}^{\top} - \boldsymbol{Z}_i^{kl}\boldsymbol{\alpha}_i\right]_t\right)\right. \\ &\quad \left. +\left(1 + \frac{[\boldsymbol{\alpha}_i]_t^2}{\left[\boldsymbol{\Sigma}_i^{-1}\right]_{t,t}}\right)\left[\boldsymbol{Z}_i^{kl}\boldsymbol{\Sigma}_i^{-1}\right]_{t,t}\right\}. \end{aligned} \tag{21}$$

The derivative (21) is easily derived based on the derivative of the log of LOO predictive distribution for single regression problem (see Chap. 5 in [10]). Throughout this paper, we refer to source separation methods based on the minimization of (16) and (20) as GPSS-ML and GPSS-LOO, respectively.

4 Implementation Issues

Our loss function (16) or (20) involves the matrix inversion ($\boldsymbol{\Sigma}_i^{-1}$), which requires high computational complexity and is often numerically unstable. As in kernel methods, we use Cholesky decomposition instead of the direct inversion of $\boldsymbol{\Sigma}_i$. For example, in (17), $\boldsymbol{\alpha}_i$ is calculated by solving the following linear systems

$$\left(\boldsymbol{L}_i\boldsymbol{L}_i^\top\right)\boldsymbol{\alpha}_i = \boldsymbol{y}_{i,1:N}^\top,$$

where $\boldsymbol{L}_i$ be the lower-triangular matrix in the Cholesky decomposition of $\boldsymbol{\Sigma}_i$. Furthermore, the log of the determinant of $\boldsymbol{\Sigma}_i$ is easily calculated through

$$\log|\det\boldsymbol{\Sigma}_i| = 2\sum_{t=1}^{N}\log\left[\boldsymbol{L}_i\right]_{t,t}.$$

A widely-used method to approximate the kernel matrix$\boldsymbol{K}_i$ is the *Nyström* method where we choose $M < N$ landmark points and use only the information in $M\times M$ submatrix $\boldsymbol{K}^{M,M}$ and $N\times M$ submatrix $\boldsymbol{K}^{N,M}$ to extrapolate elements in $\boldsymbol{K}^{N-M,N-M}$. The Nyström approximation of $\boldsymbol{K}_i$ [13], denoted by $\widetilde{\boldsymbol{K}}_i$, takes the form

$$\widetilde{\boldsymbol{K}}_i = \boldsymbol{K}_i^{N,M}(\boldsymbol{K}_i^{M,M})^{-1}\boldsymbol{K}_i^{M,N}. \tag{22}$$

In addition, other approximation methods include subset of regressor (SoR) [14], projected process (PP) [15, 10], and sparse Gaussian process (SGP) using pseudo-input [16]. A unifying view of such approximation methods is given in [17]. In this paper we only consider the marginal likelihood of $\boldsymbol{y}_{i,1:N}^\top$ (the LOO predictive distribution is not considered in approximation methods). In our case, all those approximation methods (SoR, PP, SGP, Nyström) lead to the same approximation of the marginal likelihood that is of the form

$$\begin{aligned}\log\widetilde{p}(\boldsymbol{y}_{i,1:N}|\mathcal{Y}_i) = &-\frac{1}{2}\log|\det(\widetilde{\boldsymbol{K}}_i+\boldsymbol{\Lambda}_i)|\\ &-\frac{1}{2}\boldsymbol{y}_{i,1:N}(\widetilde{\boldsymbol{K}}_i+\boldsymbol{\Lambda}_i)^{-1}\boldsymbol{y}_{i,1:N}^\top - \frac{N}{2}\log 2\pi,\end{aligned} \tag{23}$$

where $\widetilde{\boldsymbol{K}}_i$ is the approximation of $\boldsymbol{K}_i$, given in (22). Depending on approximation methods, only $\boldsymbol{\Lambda}_i$ is different. In the case of SoR, PP and Nyström, we use $\boldsymbol{\Lambda}_i = \boldsymbol{I}_N$. For SGP, $\boldsymbol{\Lambda}_i = \mathrm{diag}(\boldsymbol{K}_i - \widetilde{\boldsymbol{K}}_i) + \boldsymbol{I}_N$ for SGP. Plugging (23) into the loss function (16) leads to two different approximations: (1) GPSS-ML-Nyström (where SoR, PP, or Nyström is used for approximation); (2) GPSS-ML-SGP.

With a low-rank approximation where $\boldsymbol{K}_i \approx \widetilde{\boldsymbol{K}}_i = \boldsymbol{Q}\boldsymbol{Q}^\top$ (for instance, in Nyström method, $\boldsymbol{Q} = \boldsymbol{K}_i^{N,M}(\boldsymbol{K}_i^{M,M})^{-\frac{1}{2}}$ where $\boldsymbol{Q}\in\mathbb{R}^{N\times M}$ and $M \ll N$), the following relations are useful in saving computational load:

$$(\widetilde{\boldsymbol{K}}_i+\boldsymbol{I}_N)^{-1} = \boldsymbol{I}_N - \boldsymbol{Q}\left(\boldsymbol{I}_M+\boldsymbol{Q}^\top\boldsymbol{Q}\right)^{-1}\boldsymbol{Q}^\top, \tag{24}$$

$$\det\left(\widetilde{\boldsymbol{K}}_i+\boldsymbol{I}_N\right) = \det\left(\boldsymbol{I}_M+\boldsymbol{Q}^\top\boldsymbol{Q}\right), \tag{25}$$

where calculations are done with lower dimension M.

5 Numerical Experiments

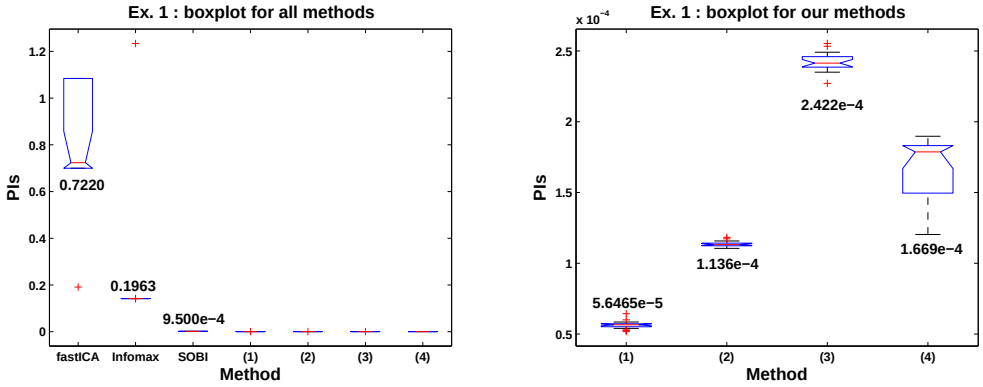
We present three empirical results with comparison to FastICA, Infomax, and SOBI, in cases where: (1) sources are nonlinear time series; (2) sources have similar spectra; (3) sources do not match the model assumptions. In all experiments, we evaluate the performance of algorithms considered here using the following performance index (PI)

$$\mathrm{PI} = \frac{1}{n}\sum_{i=1}^{n}\left\{\left(\sum_{k=1}^{n}\frac{|g_{i,k}|^2}{\max_j |g_{i,j}|^2} - 1\right) + \left(\sum_{k=1}^{n}\frac{|g_{k,i}|^2}{\max_j |g_{j,i}|^2} - 1\right)\right\}, \quad (26)$$

where $g_{i,j}$ is the (i,j)-element of the global transformation $\boldsymbol{G} = \boldsymbol{W}\boldsymbol{A}$. When perfect separation is achieved, PI=0. In practice, PI < 0.005 gives good performance. We conduct 20 independent runs for each algorithm with different initial conditions and report the statistical quantity of PI, i.e., box-plot of PI of each method and the average of value of PI. For SOBI, we use 10 different time-delayed correlation matrices to estimate the demixing matrix.

5.1 Experiment 1

We use two nonlinear time series as sources to generate $\boldsymbol{x}_t$. One source is *Santa Fe* competition laser and the other source is *Mackey-Glass* MG_{30}. In this case, our method and SOBI successfully achieve separation, while FastICA and Infomax have difficulty in separating out those two nonlinear time series (see Fig. 1).



(a) (b)

Fig. 1. Box plots of PIs (over 20 independent runs) are shown in (a), for methods which include FastICA, Infomax, SOBI, GPSS-ML (1), GPSS-LOO (2), GPSS-ML-Nyström (3), and GPSS-ML-SPR (4). PIs of only our methods, (1)-(4), are shown in (b), over a smaller dynamic range. In the case of GPSS-ML-Nyström and GPSS-ML-SPR, $M = N/10$.

In principle, length-scale hyperparameters $\{\lambda_i\}$ can also be learned. However, pre-specified values of hyperparameters provide satisfactory results as well. Exemplary empirical result is shown in Fig. 2 (a) where PI of our method is evaluated with respect to $\lambda = \lambda_1 = \cdots = \lambda_n$ varying over$[4, 100]$. The model order p determines how many past samples of source are used to calculate the covariance function. Fig. 2 (b) shows the PI of our method with respect to different values of p, where $p \geq 5$ gives quite a good performance. In the rest of experiments, we use $\lambda = 50$ and $p = 5$.

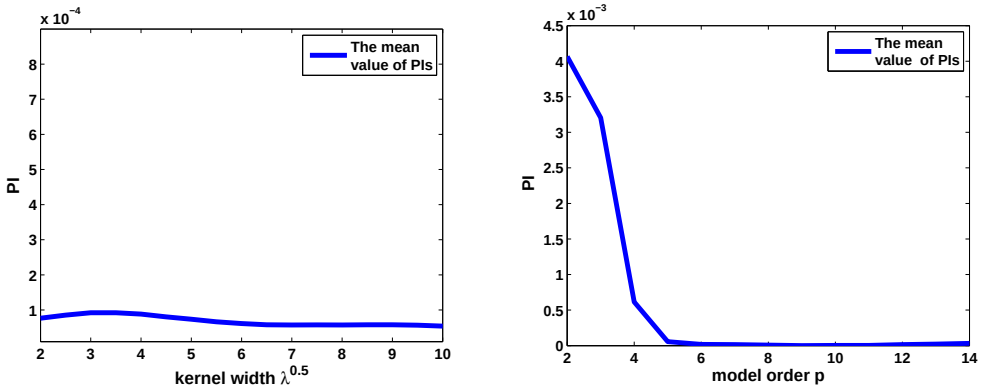


Fig. 2. The performance (in terms of PI) behavior of GPSS-ML in Experiment 1 is shown, with respect to: (a) square root of the length scale hyperparameter λ (with $p = 5$ fixed); (b) model order p (with $\lambda = 50$ fixed).

5.2 Experiment 2

We use two independent colored Gaussian sources and one music signal whose distribution is close to Gaussian to generate the observation data. Two colored Gaussian sources are generated by AR models, the coefficients of which, $\boldsymbol{h}_i \triangleq \{h_{i,\tau}\}$, are given by

$$\boldsymbol{h}_1 = \{1.3117, -0.8664, 0.5166, -0.2534\},$$
$$\boldsymbol{h}_2 = \{0.7838, 0.3988, -0.4334, -0.1792\}.$$

Certainly FastICA and Infomax do not work in this case, since sources are Gaussian. With randomly generated Gaussian innovation sequences, we chose the case where power spectra of two colored Gaussian sources are similar each other (see Fig. 3). In such a case, the performance of SOBI degrades, while our method still retains satisfactory performance (see Fig. 4-(a)). In this experiment we set $M = N/6$.

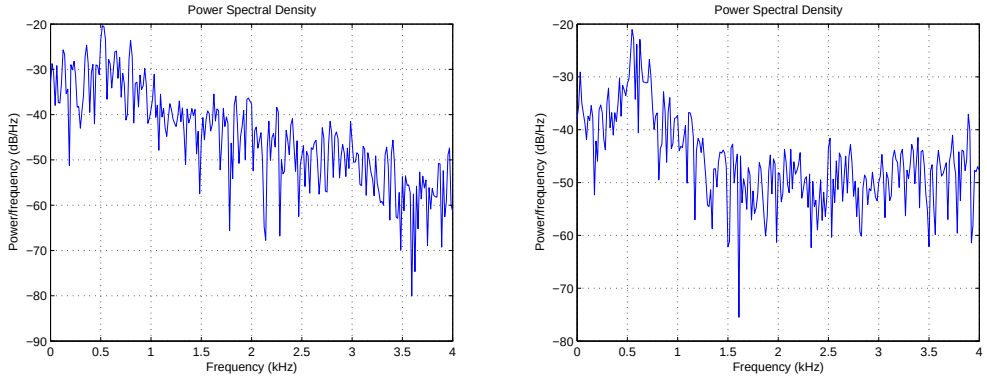
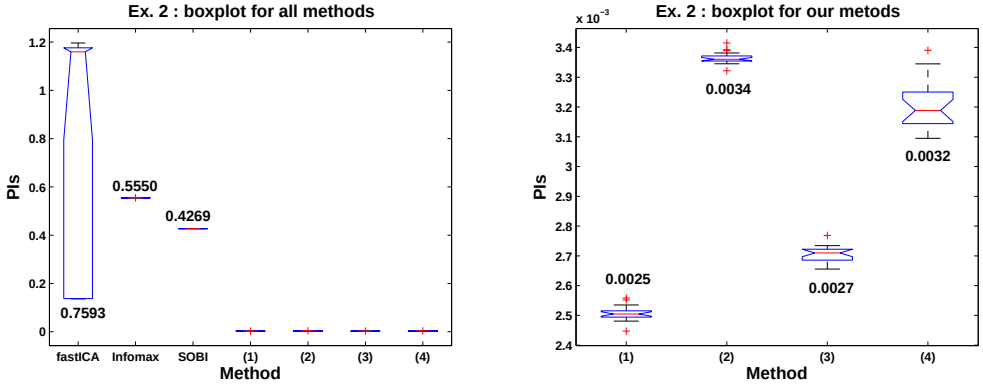


Fig. 3. Power spectrum of two synthetic colore Gaussian sources used in Experiment 2.



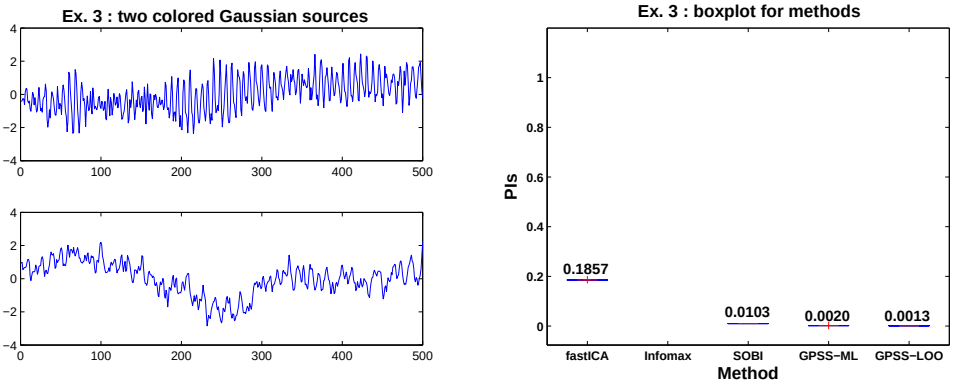
(a) (b)

Fig. 4. The box-plots of PIs are shown in the case of Experiment 2, where (1) GPSS-ML, (2) GPSS-LOO, (3) GPSS-ML-Nyström, (4) GPSS-ML-SPR.

5.3 Experiment 3

Fig. 2 (b) shows that the performance of our methods (GPSS-ML and GPSS-LOO) does not vary much in the case for overestimating p. In Experiment 3, we consider the case where we underestimate the model order p which determines how many past samples are taken into account in calculating the covariance matrix. We generate two synthetic colored Gaussian sources using linear AR models of order 20 (see Fig. 5 (a)). The case for underestimating p can be viewed as a model mis-specification. GPSS-ML and GPSS-LOO are compared in this case where $p = 5$ is used in computing $\boldsymbol{K}_i$. The marginal likelihood represents the probability of a source given the assumption of the model. In contrast, the LOO value given an estimate for the predictive distribution whether

or not the assumptions of the model may be fulfilled. In this sense Wahba has argued that the LOO-based method should be more robust against the model mis-specification [12]. In our experiment, GPSS-LOO gives slightly better performance than GPSS-ML (see Fig. 5 (b)), although the performance difference is negligible.



(a) (b)

Fig. 5. Two colored Gaussian sources generated by AR models of order 20, are shown in (a). Performance comparison is shown in (b) in the case where we underestimate the model order p in our GPSS-ML and GPSS-LOO ($p = 5$ is used). In (b), the result of Infomax is omitted since its mean value of PIs is greater than 1.

6 Conclusions

We have presented methods of source separation where we use GPs to model the temporal structure of sources and learn the demixing matrix through the mutual information minimization. The marginal likelihood of source or the LOO predictive distribution was used to model the probability distribution of source, leading to two different source separation algorithms (GPSS-ML and GPSS-LOO). Approximation methods (such as Nyström and SGP) were also used, leading to GPSS-ML-Nyström and GPSS-ML-SGP. Compared to source separation methods where a parametric method (e.g., AR model) was used to model the temporal structure of sources, our method is more flexible in the sense that: (1) sources are allowed to be nonlinear time series; (2) it is not sensitive to the model order mismatch. Compared to SOBI, our method successfully worked even in the case where sources have similar power spectra. The computational scalability in our current method is not as good as existing methods. Although we have applied several approximation methods for marginal likelihood, our method is limited to large scale data yet. This is the main drawback to be further studied, possibly adopting sparse approximations [18] that have exploited for kernel machines.

Acknowledgments: This work was supported by Korea MCIE under Brain Neuroinformatics Program and by KOSEF Basic Research Program (grant R01-2006-000-11142-0).

References

1. Cichocki, A., Amari, S.: Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications. John Wiley & Sons, Inc. (2002)
2. Bell, A., Sejnowski, T.: An information maximisation approach to blind separation and blind deconvolution. Neural Computation **7** (1995) 1129–1159
3. Hyvärinen, A., Oja, E.: A fast fixed-point algorithm for independent component analysis. Neural Computation **9** (1997) 1483–1492
4. Amari, S., Cardoso, J.F.: Blind source separation: Semiparametric statistical approach. IEEE Trans. Signal Processing **45** (1997) 2692–2700
5. Belouchrani, A., Abed-Merain, K., Cardoso, J.F., Moulines, E.: A blind source separation technique using second order statistics. IEEE Trans. Signal Processing **45** (1997) 434–444
6. Pearlmutter, B., Parra, L.: A context-sensitve generalization of ICA. In: Proceedings of International Conference on Neural Information Processing. (1996) 151–157
7. Attias, H., Schreiner, C.E.: Blind source separation and deconvolution: The dynamic component analysis algorithms. Neural Computation **10** (1998) 1373–1424
8. Cheung, Y.M.: Dual auto-regressive modelling approach to Gaussian process identification. In: Proceedings of IEEE International Conference on Multimedia and Expo. (2001) 1256–1259
9. Seeger, M.: Gaussian processes for machine learning. International Journal of Neural Systems **14** (2004) 69–106
10. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. MIT Press (2006)
11. Besag, J.: Statistical analysis of non-lattice data. The Statistician **24** (1975) 179–195
12. Wahba, G.: Spline Models for Observational Data. SIAM [Society for Industrial and Applied Mathematics] (1990)
13. Williams, C.K.I., Seeger, M.: Using the Nyström method to speed up kernel machines. In: Advances in Neural Information Processing Systems. Volume 13., MIT Press (2001)
14. Silverman, B.W.: Some aspects of the spline smoothing approach to non-parametric regression curve fitting. Journal of the Royal Statistical Society **47** (1985) 1–52
15. Seeger, M., Williams, C.K.I., Lawrence, N.D.: Fast forward selection to speed up sparse Gaussian process regression. In: Proceedings of International Workshop on Artificial Intelligence and Statistics. (2003)
16. Snelson, E., Ghahramani, Z.: Sparse Gaussian processes using pseudo-inputs. In: Advances in Neural Information Processing Systems. Volume 18., MIT Press (2006)
17. Quiñonero-Candela, J., Rasmussen, C.E.: A unifying view of sparse approximate Gaussian process regression. Journal of Machine Learning Research **6** (2005) 1939–1959
18. Csató, L., Opper, M.: Sparse on-line Gaussian processes. Neural Computation **14** (2002) 641–668