GAUSSIAN PROCESSES FOR SOURCE SEPARATION

Sunho Park and Seungjin Choi

Department of Computer Science, POSTECH, Korea
$\{$*titan,seungjin*$\}$*@postech.ac.kr*

ABSTRACT

In this paper we present a probabilistic method for source separation in the case where each source has a certain unknown temporal structure. We tackle the problem of source separation by maximum pseudo-likelihood estimation, representing the latent function which characterizes the temporal structure of each source by a random process with a Gaussian prior. The resulting pseudo-likelihood of the data is Gaussian, determined by a mixing matrix as well as by the predictive mean and covariance matrix that can be easily computed by Gaussian process (GP) regression. Gradient-based optimization is applied to estimate the demixing matrix through maximizing the log-pseudo-likelihood of the data. Numerical experiments confirm the useful behavior of our method, compared to existing source separation methods.

Index Terms— Gaussian process regression, independent component analysis, pseudo-likelihood, source separation

1. INTRODUCTION

Source separation is a fundamental problem that has wide applications in machine learning, pattern recognition, and signal processing. In the simplest form of source separation, the observation data $\boldsymbol{x}_t = [x_{1,t}, \ldots, x_{n,t}]^\top$ ($x_{i,t}$ represents the ith element of $\boldsymbol{x}_t \in \mathbb{R}^n$) is assumed to be generated by

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

where $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ is the nonsingular mixing matrix and $\boldsymbol{s}_t \in \mathbb{R}^n$ is the source vector whose elements are assumed to be statistically independent. The task of source separation 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 an ensemble of data $\{\boldsymbol{x}_t\}_{t=1}^N$. In other words, source separation aims to estimate a demixing matrix $\boldsymbol{W} = \boldsymbol{A}^{-1}$ such that $\boldsymbol{W}\boldsymbol{A} = \boldsymbol{P}\boldsymbol{\Lambda}$ where $\boldsymbol{P}$ is the permutation matrix and $\boldsymbol{\Lambda}$ is an arbitrary invertible diagonal matrix.

Various methods for source separation have been developed (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, 7]. In such cases, parameters involving AR source generative models should be also estimated in learning a mixing matrix or a demixing matrix.

Gaussian process (GP) model has been widely used in machine learning because of its flexible nonparametric nature and computational simplicity. See [8, 9] for a review and references therein. In this paper we use a Gaussian process (GP) model to characterize the temporal structure of a source, representing the postulated relationship by a distribution of latent functions. The latent function which relates the current sample of source to past samples is represented by a random process with a Gaussian prior. Integrating out latent functions is tractable, leading to a Gaussian pseudo-likelihood that is determined by a mixing matrix as well as by the predictive mean and covariance matrix that can be easily computed by GP regression. The demixing matrix is estimated by maximizing the log-pseudo-likelihood of the data. Several useful aspects of our method are summarized. The flexible nonparametric nature allows sources to be nonlinear time series and makes the method not to be sensitive to the model order reflecting how many past samples influence the current sample. Furthermore, the method achieves separation even when sources have similar power spectra, while SOBI has a difficulty in such a case.

2. GP SOURCE GENERATIVE MODEL

Incorporating the temporal structure of individual source, we model $s_{i,t}$ by

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

where $\vec{\boldsymbol{s}}_{i,t-1} \in \mathbb{R}^p$ is a collection of past p samples,

$$\vec{\boldsymbol{s}}_{i,t-1} = \left[s_{i,t-1}, s_{i,t-2}, \ldots, s_{i,t-p}\right]^\top, \tag{3}$$

and $\varepsilon_{i,t}$ is the white Gaussian noise with zero mean and unit variance, $\varepsilon_{i,t} \sim \mathcal{N}(0,1)$. The function $f_i(\cdot)$ is referred to as the *latent function* which relates the current sample $s_{i,t}$ to past samples $\vec{\boldsymbol{s}}_{i,t-1}$. In the case of linear autoregressive (AR) model, the latent function is written as

$$f_i\left(\vec{\boldsymbol{s}}_{i,t-1}\right) = \sum_{\tau=1}^{p} h_{i,\tau} s_{i,t-\tau}, \tag{4}$$

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

GP model represents the latent function $f_i(\cdot)$ by a random process with a Gaussian prior, instead of a parametric form (4). We place a GP prior over the function $f_i(\cdot)$, i.e.,

$$f_i \sim \mathcal{GP}\left(0, k(\vec{\boldsymbol{s}}_{i,t}, \vec{\boldsymbol{s}}_{i,\tau})\right), \tag{5}$$

1-4244-1484-9/08/$25.00 ©2008 IEEE

where $k\left(\vec{\boldsymbol{s}}_{i,t}, \vec{\boldsymbol{s}}_{i,\tau}\right))$ is a *covariance function*. We use the squared exponential covariance function, i.e.,

$$k(\vec{\boldsymbol{s}}_{i,t}, \vec{\boldsymbol{s}}_{i,\tau}) = \exp\left\{-\lambda_i \|\vec{\boldsymbol{s}}_{i,t} - \vec{\boldsymbol{s}}_{i,\tau}\|^2\right\}, \tag{6}$$

where λ_i is a length-scale hyperparameter.

The source generative model (2) with a GP prior (5), is referred to as *GP source generative model*, following the standard GP regression framework in which $\boldsymbol{s}_{i,1:N}^\top = [s_{i,1} \cdots s_{i,N}]^\top$ is a collection of responses and $\mathcal{S}_i = \{\vec{\boldsymbol{s}}_{i,t-1}\}_{t=1}^N$ correspond to covariates. It follows from standard GP regression that the predictive distribution of $f_i^* = f_i(\vec{\boldsymbol{s}}_{i,*})$ given $\vec{\boldsymbol{s}}_{i,*}$ is described by

$$p\left(f_i^* \mid \boldsymbol{s}_{i,1:N}^\top, \mathcal{S}_i, \vec{\boldsymbol{s}}_{i,*}\right) = \mathcal{N}\left(\bar{f}_i^*, \text{var}(f_i^*)\right), \tag{7}$$

where the mean $\bar{f}_i^*$ and variance var(f_i^*) are determined by

$$\begin{aligned} \bar{f}_i^* &= [\boldsymbol{k}_i(\vec{\boldsymbol{s}}_{i,*})]^\top \boldsymbol{K}_i^{-1} \boldsymbol{s}_{i,1:N}^\top, & (8)\\ \text{var}(f_i^*) &= k(\vec{\boldsymbol{s}}_{i,*}, \vec{\boldsymbol{s}}_{i,*}) - [\boldsymbol{k}_i(\vec{\boldsymbol{s}}_{i,*})]^\top \boldsymbol{K}_i^{-1} \boldsymbol{k}_i(\vec{\boldsymbol{s}}_{i,*}), & (9) \end{aligned}$$

where $\boldsymbol{I}$ is the identity matrix with appropriate dimension, $\boldsymbol{K}_i$ is a $N \times N$ matrix whose (u, v)-element is given by

$$[\boldsymbol{K}_i]_{u,v} = k(\vec{\boldsymbol{s}}_{i,u-1}, \vec{\boldsymbol{s}}_{i,v-1}) + \delta_{u,v}, \tag{10}$$

where $\delta_{u,v}$ is the Kronecker delta (which is 1 if $u = v$ and otherwise 0) and $\boldsymbol{k}_i(\vec{\boldsymbol{s}}_{i,*})$ is an N-dimensional vector given by

$$\boldsymbol{k}_i(\vec{\boldsymbol{s}}_{i,*}) = [k(\vec{\boldsymbol{s}}_{i,0}, \vec{\boldsymbol{s}}_{i,*}) \cdots k(\vec{\boldsymbol{s}}_{i,N-1}, \vec{\boldsymbol{s}}_{i,*})]^\top. \tag{11}$$

Taking the source generative model (2) with GP prior (5) into account, the mixing model (1) can be written as

$$\boldsymbol{x}_t = \boldsymbol{A}\boldsymbol{f}_{t-1} + \boldsymbol{A}\boldsymbol{\varepsilon}_t, \tag{12}$$

where $\boldsymbol{f}_{t-1} \in \mathbb{R}^n$ is the latent vector given by

$$\boldsymbol{f}_{t-1} = [f_{1,t-1} \cdots f_{n,t-1}]^\top, \tag{13}$$

where $f_{i,t-1} = f_i(\vec{\boldsymbol{s}}_{i,t-1})$ and $\boldsymbol{\varepsilon}_t \in \mathbb{R}^n$ follows independent Gaussian distribution, i.e., $p(\boldsymbol{\varepsilon}_t) = \mathcal{N}(0, \boldsymbol{I})$. The induced model (12) is one of key ingredients in our method, which will be used in the next section.

3. GP SOURCE SEPARATION

In maximum likelihood source separation, sources are treated as latent variables that are marginalized out. In our induced model (12) with GP prior (5), we consider the pseudo-likelihood which approximates the likelihood with a product of conditional distributions of $\boldsymbol{x}_t$ given its neighbors,

$$\text{pseudo-likelihood} = \prod_{t=1}^N p\left(\boldsymbol{x}_t | \boldsymbol{X}^{(\setminus t)}\right), \tag{14}$$

where $\boldsymbol{X}^{(\setminus t)} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{t-1}, \boldsymbol{x}_{t+1}, \ldots, \boldsymbol{x}_N\}$.

We estimate the demixing matrix $\boldsymbol{W} = \boldsymbol{A}^{-1}$ by pseudo-likelihood maximization with integrating out the latent vector $\boldsymbol{f}_{t-1}$. To this end, we consider a single factor of the pseudo-likelihood of the data

$$p(\boldsymbol{x}_t | \boldsymbol{X}^{(\setminus t)}) = \int p(\boldsymbol{x}_t | \boldsymbol{f}_{t-1}, \boldsymbol{X}^{(\setminus t)}) p(\boldsymbol{f}_{t-1} | \boldsymbol{X}^{(\setminus t)}) d\boldsymbol{f}_{t-1}, \tag{15}$$

where

$$\begin{aligned} p(\boldsymbol{x}_t | \boldsymbol{f}_{t-1}, \boldsymbol{X}^{(\setminus t)}) &= \mathcal{N}(\boldsymbol{A}\boldsymbol{f}_{t-1}, \boldsymbol{A}\boldsymbol{A}^\top), & (16)\\ p(\boldsymbol{f}_{t-1} | \boldsymbol{X}^{(\setminus t)}) &= \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t). & (17) \end{aligned}$$

The predictive mean vector $\boldsymbol{\mu}_t \in \mathbb{R}^n$ and diagonal covariance matrix $\boldsymbol{\Sigma}_t \in \mathbb{R}^{n \times n}$ are calculated using (8) and (9), which are given by

$$\begin{aligned} \mu_{i,t} &= \left[\boldsymbol{k}_i^{(\setminus t)}(\vec{\boldsymbol{s}}_{i,t-1})\right]^\top \left[\boldsymbol{K}_i^{(\setminus t)}\right]^{-1} [\boldsymbol{s}_{i,1:N}^{(\setminus t)}]^\top, & (18)\\ \boldsymbol{\Sigma}_t &= \text{diag}\left(\sigma_{1,t}^2, \ldots, \sigma_{n,t}^2\right), & (19)\\ \sigma_{i,t}^2 &= k(\vec{\boldsymbol{s}}_{i,t-1}, \vec{\boldsymbol{s}}_{i,t-1}) \\ &\quad - \left[\boldsymbol{k}_i^{(\setminus t)}(\vec{\boldsymbol{s}}_{i,t-1})\right]^\top \left[\boldsymbol{K}_i^{(\setminus t)}\right]^{-1} \boldsymbol{k}_i^{(\setminus t)}(\vec{\boldsymbol{s}}_{i,t-1}), & (20) \end{aligned}$$

where $\boldsymbol{k}_i^{(\setminus t)}(\vec{\boldsymbol{s}}_{i,t-1})$ and $\boldsymbol{s}_{i,1:N}^{(\setminus t)}$ are $(N-1)$-dimensional vectors where the element t is eliminated

$$\begin{aligned} \boldsymbol{k}_i^{(\setminus t)}(\vec{\boldsymbol{s}}_{i,t-1}) &= [k(\vec{\boldsymbol{s}}_{i,0}, \vec{\boldsymbol{s}}_{i,t-1}), \ldots, k(\vec{\boldsymbol{s}}_{i,t-2}, \vec{\boldsymbol{s}}_{i,t-1}), \\ &\quad\; k(\vec{\boldsymbol{s}}_{i,t}, \vec{\boldsymbol{s}}_{i,t-1}), \ldots, k(\vec{\boldsymbol{s}}_{i,N-1}, \vec{\boldsymbol{s}}_{i,t-1})]^\top, \\ \boldsymbol{s}_{i,1:N}^{(\setminus t)} &= [s_{i,1}, \ldots, s_{i,t-1}, s_{i,t+1}, \ldots, s_{i,N}], \end{aligned}$$

and $\boldsymbol{K}_i^{(\setminus t)} \in \mathbb{R}^{(N-1)\times(N-1)}$ is a submatrix of $\boldsymbol{K}_i$ where column t and row t are removed.

Mean $\mu_{i,t}$ and variance $\sigma_{i,t}^2$ are efficiently computed from the inverse of the complete covariance matrix using inversion by partitioning (see Ch. 5 in [9] or [10]), leading to

$$\begin{aligned} \mu_{i,t} &= s_{i,t} - \frac{\left[\boldsymbol{K}_i \boldsymbol{s}_{i,1:N}^\top\right]_t}{\left[\boldsymbol{K}_i^{-1}\right]_{t,t}}, & (21)\\ \sigma_{i,t}^2 &= \frac{1}{\left[\boldsymbol{K}_i^{-1}\right]_{t,t}} - \text{var}(\varepsilon_{i,t}) = \frac{1}{\left[\boldsymbol{K}_i^{-1}\right]_{t,t}} - 1. & (22) \end{aligned}$$

It follows from (15), (16), and (17) that the single factor of the pseudo-likelihood follows Gaussian distribution,

$$p(\boldsymbol{x}_t | \boldsymbol{X}^{(\setminus t)}) = \mathcal{N}(\boldsymbol{A}\boldsymbol{\mu}_t, \boldsymbol{\Gamma}_t), \tag{23}$$

where $\boldsymbol{\Gamma}_t = \boldsymbol{A}(\boldsymbol{\Sigma}_t + \boldsymbol{I})\boldsymbol{A}^\top$. Thus we write the log-pseudo-likelihood of the data as

$$\begin{aligned} \mathcal{L} &= \sum_{t=1}^N \log p\left(\boldsymbol{x}_t | \boldsymbol{X}^{(\setminus t)}\right) \\ &= -\frac{1}{2} \sum_{t=1}^N \left\{\log 2\pi + \log |\boldsymbol{\Gamma}_t| + \boldsymbol{\beta}_t^\top (\boldsymbol{\Sigma}_t + \boldsymbol{I}) \boldsymbol{\beta}_t\right\}, & (24) \end{aligned}$$

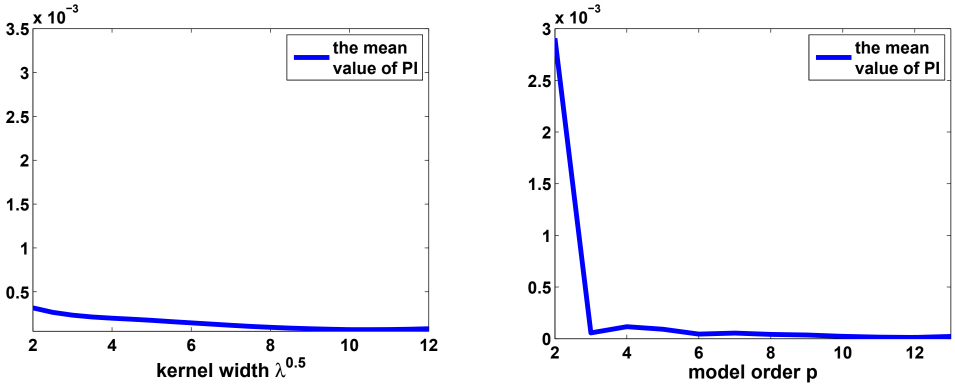
where $[\boldsymbol{\beta}_t]_i = \left[\boldsymbol{K}_i^{-1} \boldsymbol{s}_{i,1:N}^\top\right]_t$ and relations (21) and (22) are used to derive the last equality.

We estimate the demixing matrix $\boldsymbol{W}$ by maximizing the log-pseudo-likelihood (24). A gradient-based optimization is applied to find a solution which maximize (24), where `fminunc` in Matlab optimization toolbox [11] was used in our implementation. In order to compute the gradient, we first define

$$\boldsymbol{Z}_i^{kl} = \boldsymbol{K}_i^{-1} \frac{\partial \boldsymbol{K}_i}{\partial w_{k,l}} \boldsymbol{K}_i^{-1}, \tag{25}$$

where the derivative of the covariance matrix $\boldsymbol{K}_i$ with respect to $w_{k,l}$ (which is the (k, l)-element of the demixing matrix $\boldsymbol{W}$) is computed as

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



(a)

(b)

Fig. 1. Two nonlinear time series sources (Mackey-Glass MG_{30} and Santa Fe competition Laser) are used to generate the mixture. The performance evaluation of our method is shown, with respect to: (a) the square root of the length-scale hyperparameter λ (with $p = 5$ fixed); (b) model order p (with $\lambda = 50$ fixed).

where $\boldsymbol{\Delta} \triangleq [(\boldsymbol{x}_{u-1} - \boldsymbol{x}_{v-1}), \ldots, (\boldsymbol{x}_{u-p} - \boldsymbol{x}_{v-p})]$. With this definition, the gradient of (24) with respect to $w_{k,l}$ is determined by

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial w_{k,l}} &= -\frac{1}{2}\sum_{t=1}^{N} \mathrm{tr}\left\{\boldsymbol{\Gamma}_t^{-1}\frac{\partial}{\partial w_{k,l}}\left[\boldsymbol{W}^{-1}(\boldsymbol{\Sigma}_t+\boldsymbol{I})\boldsymbol{W}^{-\top}\right]\right. \\ &\quad + \left(\frac{\partial}{\partial \boldsymbol{\beta}_t}\left[\boldsymbol{\beta}_t^{\top}(\boldsymbol{\Sigma}_t+\boldsymbol{I})\boldsymbol{\beta}_t\right]\right)^{\top}\frac{\partial \boldsymbol{\beta}_t}{\partial w_{k,l}} \\ &\quad \left. + \left(\frac{\partial}{\partial \boldsymbol{\Sigma}_t}\left[\boldsymbol{\beta}_t^{\top}(\boldsymbol{\Sigma}_t+\boldsymbol{I})\boldsymbol{\beta}_t\right]\right)^{\top}\frac{\partial \boldsymbol{\Sigma}_t}{\partial w_{k,l}}\right\},\end{aligned}$$

which is calculated as

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial w_{k,l}} &= \frac{1}{2}\sum_{t=1}^{N} \mathrm{tr}\left\{2\boldsymbol{W}^{-1}\frac{\partial \boldsymbol{W}}{\partial w_{k,l}} - \boldsymbol{\Gamma}_t^{-1}\boldsymbol{W}^{-1}\frac{\partial \boldsymbol{\Sigma}_t}{\partial w_{k,l}}\boldsymbol{W}^{-\top}\right. \\ &\quad \left. -2\boldsymbol{\beta}_t^{\top}(\boldsymbol{\Sigma}_t+\boldsymbol{I})\frac{\partial \boldsymbol{\beta}_t}{\partial w_{k,l}} - \boldsymbol{\beta}_t^{\top}\frac{\partial \boldsymbol{\Sigma}_t}{\partial w_{k,l}}\boldsymbol{\beta}_t\right\}, \qquad (26)\end{aligned}$$

where

$$\frac{\partial \boldsymbol{\beta}_t}{\partial w_{k,l}} = \left[-\boldsymbol{Z}_i^{kl}\boldsymbol{s}_{i,1:N}^{\top} + \boldsymbol{K}_i^{-1}\boldsymbol{x}_{l,1:N}^{\top}\right]_t \delta_{i,k}, \qquad (27)$$

$$\left[\frac{\partial \boldsymbol{\Sigma}_t}{\partial w_{k,l}}\right]_{i,i} = \frac{\left[\boldsymbol{Z}_i^{kl}\right]_{t,t}}{\left[\boldsymbol{K}_i^{-1}\right]_{t,t}^{2}}\delta_{i,k}. \qquad (28)$$

The hyperparameters λ_i can be also learned by maximum pseudo-likelihood, since the gradient with respect to them is easily computed. However, here we fix them as constant values and learn the demixing matrix only. One of empirical results show that the performance does not much depend on the values of hyperparameters (see Fig. 1 (a)).

4. NUMERICAL EXPERIMENTS

We compare our GP source separation method to several existing methods such as FastICA [3], Infomax [2, 12], SOBI [5], and dual AR model-based method [7] (that is referred to as AR-BSS). FastICA and Infomax exploit only non-Gaussianity of sources, without considering any their temporal structure. SOBI is a algebraic method which jointly diagonalizes a set of time-delayed covariance matrices to estimate the demixing matrix. The dual AR model-based method uses linear AR models to take the temporal structure of source into account.

We present two empirical results in cases where: (1) sources have similar spectra; (2) sources are nonlinear time series. We evaluate the performance of algorithms using the performance index (PI) (also known as Amari index [12]) defined by

$$\begin{aligned}\mathrm{PI} &= \frac{1}{n}\sum_{i=1}^{n}\left\{\left(\sum_{k=1}^{n}\frac{|g_{i,k}|^2}{\max_j |g_{i,j}|^2} - 1\right)\right. \\ &\quad \left. + \left(\sum_{k=1}^{n}\frac{|g_{k,i}|^2}{\max_j |g_{j,i}|^2} - 1\right)\right\}, \qquad (29)\end{aligned}$$

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 and PI < 0.05 provides reasonable performance. We conduct 20 independent runs for each algorithm with different initial conditions and report the average value of PI.

Throughout experiments, we fix length-scale hyperparameters as $\lambda_i = 50$ for $i = 1, \ldots, n$. In principle, such hyperparameters can be also learned through maximum pseudo-likelihood estimation. However, pre-specified values of hyperparameters do not much influence the final performance in our case. Exemplary empirical result is shown in Fig. 1 (a), where PI is evaluated with $\lambda = \lambda_1 = \lambda_2$ varying over $[2^2, 12^2]$ in the case where the observed data is a mixture of two nonlinear time series including *Mackey-Glass* MG_{30} and *Santa Fe competition Laser*. We also fix the model order p as $p = 5$. Exemplary performance with p varying from 1 to 13 is shown in Fig. 1 (a), where the same two nonlinear time series are used as sources.

4.1. Experiment 1

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 of order $p = 4$. Certainly FastICA and Infomax do not work in this case, since sources are Gaussian. In the case where power spectra of two colored Gaussian sources are similar each other, the performance of SOBI degrades, while our method and AR-BSS still retains satisfactory performance (see Fig. 2).

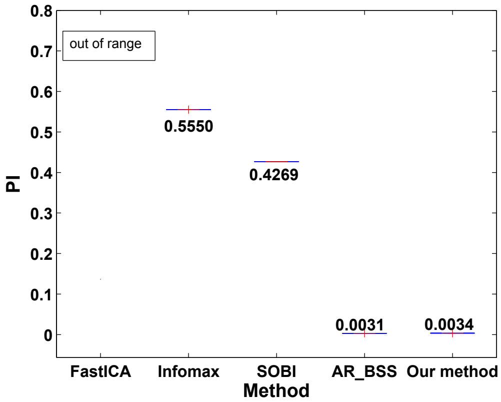


Fig. 2. The performance comparison in Experiment1. ”Out of range” means that PI in FastICA is over 0.8 (the maximum scale in this plot). AR-BSS and our method work successfully in achieving the separation of Gaussian sources.

4.2. Experiment 2

In this experiment we use nonlinear time series (Mackey-Glass MG_{30}, Santa Fe competition laser, the first variable Chaotic data Ikeda map) as sources. Our method shows the best performance, compared to SOBI and AR-BSS (see Fig. 3), although the performance difference is not so big. SOBI does not assume any linear temporal modeling since it exploits only time-delayed covariance structure. Linear AR modeling seems to be fine even in the case where actual sources are nonlinear times series. However the performance is degraded, compared to our nonparametric source modeling.

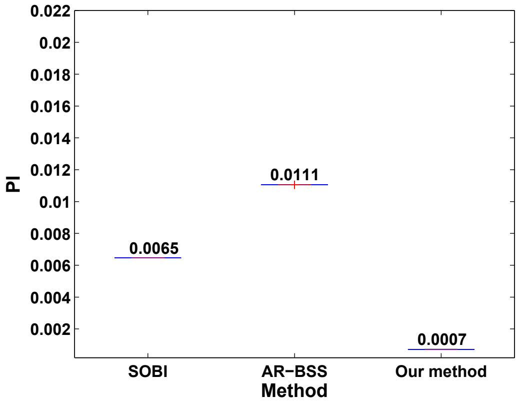


Fig. 3. The performance comparison in Experiment 2. FastICA and Infomax are omitted since their PI values are over 0.4. Our method shows the best performance, while SOBI and AR-BSS also show reasonable performance.

5. CONCLUSIONS

We have presented a source separation method where each source is modeled by a GP and the demixing matrix is learned by maximizing the log-pseudo-likelihood of the data. Compared to source separation methods where a parametric modeling (e.g., AR model) was used to capture the temporal structure of sources, our method is more flexible in the sense that: (1) sources are allowed to be nonlinear time series; (2) source generative model is not sensitive to the model order. We compared our method to two representative methods (SOBI and AR-BSS) which also exploited the temporal structure of source. Compared to SOBI, our method successfully worked even in the case where sources have similar power spectra, whereas SOBI failed. The marginal likelihood was also considered for source separation with GP models [13].

Acknowledgments: This work was supported by Korea MICE Brain Neuroinformatics program and Korea MIC under ITRC support program supervised by IITA (IITA-2007-C1090-0701-0045).

6. REFERENCES

[1] A. Cichocki and S. Amari, *Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications*, John Wiley & Sons, Inc., 2002.

[2] A. Bell and T. Sejnowski, “An information maximisation approach to blind separation and blind deconvolution,” *Neural Computation*, vol. 7, pp. 1129–1159, 1995.

[3] A. Hyvärinen and E. Oja, “A fast fixed-point algorithm for independent component analysis,” *Neural Computation*, vol. 9, pp. 1483–1492, 1997.

[4] S. Amari and J. F. Cardoso, “Blind source separation: Semiparametric statistical approach,” *IEEE Trans. Signal Processing*, vol. 45, pp. 2692–2700, 1997.

[5] A. Belouchrani, K. Abed-Merain, J. -F. Cardoso, and E. Moulines, “A blind source separation technique using second order statistics,” *IEEE Trans. Signal Processing*, vol. 45, pp. 434–444, Feb. 1997.

[6] H. Attias and C. E. Schreiner, “Blind source separation and deconvolution: The dynamic component analysis algorithms,” *Neural Computation*, vol. 10, pp. 1373–1424, 1998.

[7] Y. M. Cheung, “Dual auto-regressive modelling approach to Gaussian process identification,” in *Proceedings of IEEE International Conference on Multimedia and Expo*, 2001, pp. 1256–1259.

[8] M. Seeger, “Gaussian processes for machine learning,” *International Journal of Neural Systems*, vol. 14, no. 2, pp. 69–106, 2004.

[9] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, MIT Press, 2006.

[10] S. Sundararajan and S. S. Keerthi, “Predictive approaches for choosing hyperparameters in Gaussian processes,” *Neural Computation*, vol. 13, pp. 1103–1118, 2001.

[11] T. Coleman, M. Branch, and A. Grace, “Optimization toolbox for use with MATLAB user’s guide version 2,” 1999.

[12] S. Amari, A. Cichocki, and H. H. Yang, “A new learning algorithm for blind signal separation,” in *Advances in Neural Information Processing Systems*, D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, Eds. 1996, vol. 8, pp. 757–763, MIT Press.

[13] S. Park and S. Choi, “Source separation with Gaussian process models,” in *Proceedings of the European Conference on Machine Learning*, Warsaw, Poland, 2007, pp. 262–273, Springer.