International Symposium on Nonlinear Theory and its Applications
Xi'an, PRC, October 7–11, 2002

Source Decomposition of Superimposed Dynamical Signals

Toshihiro Iwamoto[†] and Kazuyuki Aihara[‡]

The University of Tokyo[†]
The University of Tokyo and CREST, JST, Japan[‡]
Engr. 6th bldg., 7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan
Phone:+81-3-5841-6913, Fax:+81-3-5841-8594
Email: iwamoto@sat.t.u-tokyo.ac.jp, aihara@sat.t.u-tokyo.ac.jp

1. Introduction

In many real world problems, essentially important signals cannot be directly observed, and they are only available as mixtures that include other signals or noise. To cope with this difficult situation, several methods have been developed. ICA[1] can recover original signals from linear mixtures of them by assuming independence of original signals and that the same number of mixed signals as that original signals can be observed.

However, observation of an enough number of mixture signals is usually difficult or impossible. There is an extension of ICA[2] that can cope with such situation by assuming that signals have heavy-tailed distributions. Due to the assumption of heavy-tailed distribution, however, its applicability is limited.

In this paper, we consider signals in a delay coordinate space to recover original signals from single channel observation. The reason we choose a delay coordinate approach is that the delay coordinate representation is very natural for dynamical signals. We don't use frequency domain representations because, while they usually lead to sparse data representations, they blur dynamical aspect of signals. The present method is rather close to dynamical signal noise reduction based on local projection to attractor manifold[3].

Our method especially works well for signals that are generated from low dimensional dynamical systems. While this method is not very suited for noise reduction, it works at least as good as simple linear filtering. Many real world signals, such as speech and instrumental sounds, have more or less dynamical aspects, and it should be worthwhile to apply this method to them.

In the section 2, we will describe the general framework of this method, including the probabilistic model used (and estimation framework). Detailed computation algorithms will be given in the section 3. We will show simulation results in the section 4, and give conclusion and future problems in the section 5.

2. Our signal decomposition method

Let $x_i(t)$ $(i = 1, \ldots, n_{\mathrm{sig}})$ $(t = \ldots, -1, 0, 1, 2, \ldots)$ be the original signals to recover and $y(t)$ be the mixture signal observed. We assume that $x_i(t)$ and $x_j(t)$ are independent for every $i \neq j$.

Let $P_i(\cdot)$ be the probability density of $x_i(t)$. We assume $x_i(t)$ is $(d-1)$-th order Markov process and approximate $P_i(x_i(t))$ as follows. As we are considering dynamic signals as a primal concern, this assumption is reasonable and any d twice larger than the attractor dimension would suffice.

To describe $P_i(x(t)|x(t-1), x(t-2), \ldots, x(t-d+1))$, we use the gaussian mixture model.

$$\begin{aligned}
& P_i(x(t), x(t-1) \ldots, x(t-d+1)) \\
= \; & P_i(\mathbf{x}) \\
= \; & \sum_j \frac{1}{M} \frac{1}{\sqrt{(2\pi)^d |\Sigma_{ij}|}} \exp\{-\frac{1}{2}(\mathbf{x}-\mu_{ij})^T \Sigma_{ij}^{-1} (\mathbf{x}-\mu_{ij})\},
\end{aligned} \tag{1}$$

where M is the number of the components and Σ_{ij} and μ_{ij} are the model parameters. We define $P_i(x(t)|x(t-1), x(t-2), \ldots, x(t-d+1))$ as follows.

$$\begin{aligned}
& P_i(x(t)|x(t-1) \ldots, x(t-d+1)) \\
= \; & \frac{P_i(x(t)|x(t-1), \ldots, x(t-d+1))}{P_i(x(t-1), \ldots, x(t-d+1))} \\
= \; & \frac{P_i(x(t)|x(t-1), \ldots, x(t-d+1))}{\int P_i(x, x(t-1), \ldots, x(t-d+1)) dx}
\end{aligned} \tag{2}$$

For $m > d$ we have:

$$\begin{aligned}
& \log P_i(x(t), x(t-1), \ldots, x(t-m+1)) \\
= \; & \sum_{i=0}^{m-d-1} \log P_i(x(t-i)|x(t-i-1), \ldots, \\
& \qquad x(t-i-d+1)) \\
& + \log P_i(x(t-m+d) \ldots, x(t-m+1))
\end{aligned} \tag{3}$$

Strictly speaking, eq.(1) isn't a Markov process model because the equation

$$\int P_i(x, x(t-1)\ldots, x(t-d+1))dx = \int P_i(x(t-1)\ldots, x(t-d+1), x)dx \quad (4)$$

doesn't hold in general. As eq.(3) is built upon eq.(1), eq.(3) has the same problem. Although solving this issue might make the overall estimation performance better, we won't discuss this problem further in this paper.

We assume separate training data for each of $x_1, \ldots, x_{n_{\mathrm{sig}}}$ are available. We first fit the Gaussian mixture model parameters using these data. Given a mixture time series signal $y(t)$, we estimate original signals as:

$$(\hat{\mathbf{x}}_1(t), \ldots, \hat{\mathbf{x}}_{n_{\mathrm{sig}}}(t)) = \arg\max_{\hat{\mathbf{x}}_1(t),\ldots,\hat{\mathbf{x}}_{n_{\mathrm{sig}}}(t)} \left\{ \sum_i \log P_i(\hat{\mathbf{x}}_i(t)) \right\}, \quad (5)$$

s.t. $\sum_i x_i(t) = y(t)$.

We are not sure if the maximization in eq.(2) is optimal. But as the true solution $x_i(t)$ has high log-probability, this probability maximization isn't bad if there are no false solutions.

3. Estimation Algorithm

3.1. Fitting probability model parameters

The models are fit using a standard EM algorithm. The only exception to the standard procedure is that each Σ_i is rounded so that all of its eigenvalues are no less than 10^{-5}. This trick was needed to avoid numerical instability.

E-step

$$z_{ij} \leftarrow \frac{\frac{1}{\sqrt{(2\pi)^d|\Sigma_j|}} \exp\{-(\mathbf{x}_i-\mu_j)^T\Sigma_i^{-1}(\mathbf{x}_i-\mu_j)\}/2}{\sum_{j=1}^{M}\frac{1}{\sqrt{(2\pi)^d|\Sigma_j|}} \exp\{-(\mathbf{x}_i-\mu_j)^T\Sigma_i^{-1}(\mathbf{x}_i-\mu_j)\}/2}$$

M-step

$$\mu_j \leftarrow \frac{\sum_i z_{ij}\mathbf{x}_i}{\sum_i z_{ij}}, \quad \Sigma_j \leftarrow \frac{\sum_i z_{ij}\mathbf{x}_i\mathbf{x}_i^T}{\sum_i z_{ij}}$$

3.2. Signal separation

The estimation of the original time series $x_i(t)$ is performed by solving eq.(2). This is a difficult optimization problem. We solve this by dividing it to several calculation steps.

In this section, $\hat{x_1}(t), \ldots, x_{\hat{n_{\mathrm{sig}}}}(t)$ are always chosen to satisfy $\sum_i \hat{x_i}(t) = y(t)$, thus we omit writing this constraint. For simplicity, we let $\hat{x_i}(t)$ denote $\hat{x_1}(t), \ldots, x_{\hat{n_{\mathrm{sig}}}}(t)$. t_c, t_d and t_e are the constants which controls the number of optimization calculation steps.

step A (the first estimation):
Used when there are no estimates $\hat{x_i}(t)$. The following maximization is possible using an exhaustive search.

$$\{\hat{x_i}(0), \ldots, \hat{x_i}(d-1)\}_{i=1}^{n_{\mathrm{sig}}} \leftarrow \arg\max_{\{\hat{x_i}(0),\ldots,\hat{x_i}(d-1)\}_i} \left\{ \sum_i \log P_i(\hat{x_i}(d-1), \ldots, \hat{x_i}(0)) \right\} \quad (6)$$

step B (initial estimation):
Assume $\hat{x_i}(0), \ldots, \hat{x_i}(t-1)$ are already estimated.

$$\hat{x_1}(t), \ldots, x_{\hat{n_{\mathrm{sig}}}}(t) \leftarrow \arg\max_{\{\hat{x_i}(t)\}_i} \left\{ \sum_i \log P_i(\hat{x_i}(t), \ldots, \hat{x_i}(t-d+1)) \right\} (7)$$

step C (forward update):
Assume $\hat{x_i}(0), \ldots, \hat{x_i}(t)$ are already estimated. $\hat{x_i}(t-t_c), \ldots, \hat{x_i}(t)$ are updated in the following way.

1. quasi-Newton
   Maximize $\{\sum_i \log P_i(\hat{x_i}(t-t_c-d+1), \ldots, \hat{x_i}(t))\}$ using the quasi-Newton algorithm with $\hat{x_i}(t-t_c-d+1), \ldots, \hat{x_i}(t-t_c-1)$ fixed.

2. randomize + quasi-Newton
   Add random fluctuation to $\hat{x_i}(t-t_c), \ldots, \hat{x_i}(t+i)$ and maximize $\{\sum_i \log P_i(\hat{x_i}(t-t_c-d+1), \ldots, \hat{x_i}(t)\}$ using the quasi-Newton algorithm with $\hat{x_i}(t-t_c-d+1), \ldots, \hat{x_i}(t-t_c-1)$ fixed. Use the newly calculated values if the log-probability increased, otherwise discard them and retain the old values before adding fluctuations.

step D (restart):
Assume $\hat{x_i}(0), \ldots, \hat{x_i}(t)$ are already estimated.

1. Calculate another set of estimates $\hat{x_i}'(t-t_d), \ldots, \hat{x_i}'(t)$ $(i = 1, \ldots, n_{\mathrm{sig}})$ without using $\hat{x_i}(0), \ldots, \hat{x_i}(t)$. This is done by performing step A, B and C on the sequence $y(t-t_d), \ldots, y(t)$.

2. Maximize $\{\sum_i \log P_i(\hat{x_i}(t-t_e), \ldots, \hat{x_i}(t))\}$ with $\hat{x_i}(t-t_e), \ldots, \hat{x_i}(t-t_e+d-2)$ fixed to $\hat{x_i}(t-t_e), \ldots, \hat{x_i}(t-t_e+d-2)$ and $\hat{x_i}(t-t_d), \ldots, \hat{x_i}(t)$ fixed to $\hat{x_i}'(t-t_d), \ldots, \hat{x_i}'(t)$.

3. Replace the old estimates with $\hat{x_i}'(t-t_e), \ldots, \hat{x_i}'(t)$ if it gains log-probability.

The above calculation steps are used in the following order.

1. Step A calculates $\hat{x_i}(0), \ldots, \hat{x_i}(d-1)$. Set $t = d-1$.
2. Increment t by one and calculate $\hat{x_i}(t)$ by applying Step B.
3. Step C is performed on $\hat{x_i}(t-t_c-d+1), \ldots, \hat{x_i}(t)$, if $t \geq t_c + d - 1$.
4. Step D is performed on $\hat{x_i}(t-t_e), \ldots, \hat{x_i}(t)$, if $t \geq t_e$.
5. Repeat from the step 2.

Without the step D, the error gets amplified and we cannot obtain acceptable solutions. As the step D is computationally heavy, we can reduce the numbers of step D at the cost of less optimal solution, say, by omitting the step D at every two iteration loops.

4. Simulation

We applied the source separation algorithm described in the previous section to several sets of computer generated time series to demonstrate its performance.

We used the following sets of time series.

- Rossler $(a = b = 0.2, c = 4.7)$ + Rossler $(a = 0.4, b = 0.7, c = 4)$
- Rossler $(a = 0.4, b = 0.7, c = 4)$ + Lorenz $(a = 10, b = 28, c = 8/3)$
- Rossler $(a = b = 0.2, c = 4.7)$ + Rossler $(a = 0.4, b = 0.7, c = 4)$ + Lorenz $(a = 10, b = 28, c = 8/3)$

We refer to these original time series in each set as signal 1, 2 and 3, respectively.

The equations for the Rossler system and the Lorenz system are:

$$\frac{dx}{dt} = \alpha_r(-y - z) \quad (8)$$

$$\frac{dy}{dt} = \alpha_r(x + ay) \quad (9)$$

$$\frac{dz}{dt} = \alpha_r(b + z(x - c)) \quad (10)$$

and

$$\frac{dx}{dt} = \alpha_l(a(y - x)) \quad (11)$$

$$\frac{dy}{dt} = \alpha_l(bx - y - xz) \quad (12)$$

$$\frac{dz}{dt} = \alpha_l(xy - cz), \quad (13)$$

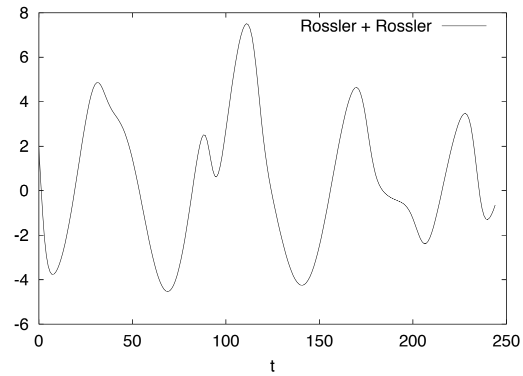

Figure 1: Example of a mixed signal

Table 1: Mean square errors of signal separation

	error	
Rossler + Rossler	0.175	
Rossler + Lorenz	0.050	
Rossler + Rossler + Lorenz	2.631	(signal 1)
	4.120	(signal 2)
	2.578	(signal 3)

respectively. To make the problems non-trivial for simple linear filtering, we set $\alpha_r = 0.1$ and $\alpha_l = 0.02$ so that these two systems have similar power spectra. These equations are numerically integrated using the Runge-Kutta method with the time interval $t = 1$, and only the values of x at $t = 0, 1, 2, \ldots$ are considered.

The signals are embedded in a 5 dimensional delay coordinate space, and a Gaussian mixture model with 10 components was used for each signals, i.e. $d = 5$ and $M = 10$. The sizes of data used for probability density estimation were 100000 for each signals.

Figure1 and fig.3 show an example of a mixed signal and its decomposition for the Rossler + Rossler case. Figure2 and fig.4 show an example of a mixed signal and its decompositions for the Rossler + Rossler + Lorenz case. The mean square errors are shown in Table1. As the mean squares of original signals are roughly from 10 to 70, we have roughly 20dB suppression of unwanted signals in the two signals problem, and 10dB in the three signals problem.

The signal separation was very successful for problems with two original signals. While the result of the three signals problem wasn't as good as those of the two signals problems, solving this problem with larger d will make the results better.

5. Conclusion

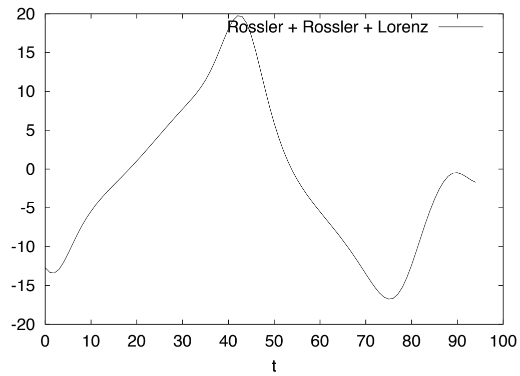

Figure 2: Example of a mixed signal

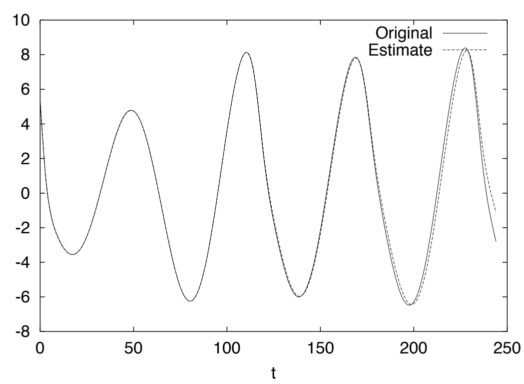

Figure 3: An original signal and its estimate

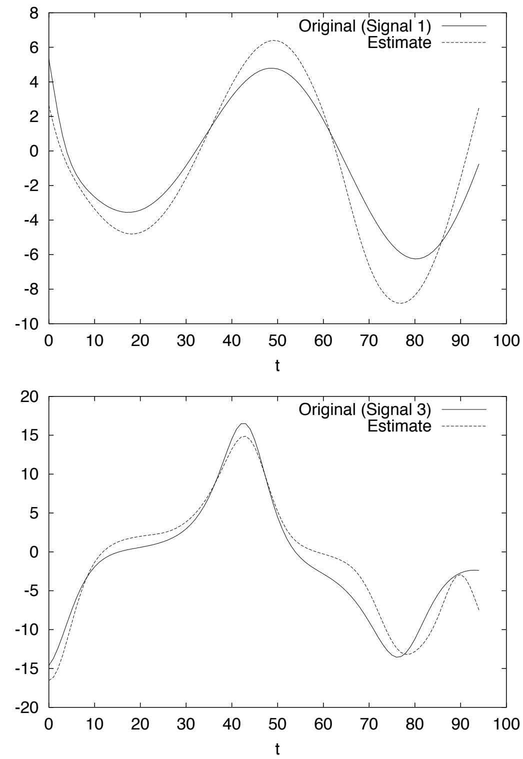

Figure 4: Original signals and their estimates

We have proposed a novel method to separate dynamical signals from only one mixture time series. Simulation results show that the proposed method is promising. Future works include application to real world complex signals.

References

[1] P. Comon, "Independent component analysis, a new concept," Signal Processing, Vol. 36, No. 3, pp. 287–314, 1994.

[2] T.-W. Lee, M. S. Lewicki, M. Girolami, and T. J. Sejnowski, "Blind source separation of more sources than mixtures using overcomplete representations," IEEE Signal Processing Letters, Vol. 6, No. 4, pp. 87–90, 1999.

[3] R. Cawley and G.-H. Hsu, "Local-geometric-projection method for noise reduction in chaotic maps and flows," Physical Review A, Vol. 46, No. 6, pp. 3057–3082, 1992.