BLIND SOURCE SEPARATION FOR TIME-VARIANT MIXING SYSTEMS USING PIECEWISE LINEAR APPROXIMATIONS

Ramon E. Prieto

Toyota InfoTechnology Center U.S.A.
Human Machine Interaction Group
rprieto@us.toyota-itc.com

Pamornpol Jinachitra

CCRMA
Stanford University
pj97@stanford.edu

ABSTRACT

In this paper we address the problem of Blind Source Separation when the mixing system is time-variant. We address the problem using piecewise linearly variant unmixing matrices in time as an approximation. Using optimization theory we are able to transform the Independent Component Analysis problem to handle linearly time-variant unmixing matrices. Using a time-adaptive whitening pre-processing step, ICA in the lie-group domain, and an optimization problem in the original unmixing matrix domain we are able to get accurate results for synthetically generated signals, bringing significant improvements over time-invariant ICA methods.

1. INTRODUCTION

A lot of work has been done in separating acoustic sound sources mixed in real room environments ([1], [2]). Three different families of algorithms exist: algorithms that work in the time domain ([3]), algorithms that work exclusively in the frequency domain ([4]), and algorithms that work in both the time domain and frequency domain at the same time ([5]).

While each family of algorithms has it's advantages and disadvantages, all of them require the acoustics of the problem to be time-invariant, meaning that the position of the sound sources and the position of the microphones must be time-invariant. This constraint might be too restrictive, for example, in the case of automobiles, where the noise sound sources might be moving at the same time the driver is speaking to it's car's far-field cell phone microphone

To address the problem of moving sound sources, a number of online algorithms have been developed. In the case of [6], they developed an algorithm that applies ICA frame by frame, in the frequency domain, and then apply a post processing stage using crosstalk component estimation and spectral subtraction to compensate for the mixing remainings. In the case of [7], they use an online PCA algorithm (SIPEX-G) to calculate the whitening matrix and an online algorithm to calculate the rotation matrix. In the work done in [8] they use a framewise on-line algorithm in the time domain using Maximum Likelihood Estimation.

We can see that, in most cases, these online algorithms are in charge of calculating unmixing matrices at each frame, independently from the other frames. In real life, the unmixing matrix might be varying even in the period of time covered by the frame, so a framewise approximation might not be optimal. Also, if the sources are non-stationary, the unmixing matrices found in the algorithm might try to compensate for the temporal characteristics of the source signals at some particular frames.

In this paper we address the problem of moving sources making a piecewise linear approximation to the trajectory made by the unmixing matrices in time. By making a piecewise linear approximation we avoid the problems mentioned above. The temporal characteristics of the signal won't influence individual frames, instead, the entire mixtures will contribute in the update of all the linear pieces that make up the time-variant system.

To achieve these goals, we first mathematically analyze the optimization problem of blind source separation for moving sources when using piecewise linear approximations of the unmixing matrices. We do this in section 2. In section 2 we also derive a gradient ascent algorithm that solves that optimization problem. An approximate whitening stage is developed in section 3 to approximate the problem to a piecewise linear rotation sequence of unmixing matrices. The piecewise linear rotation is analyzed in the lie group domain in section 4. Finally a four steps algorithm is developed in section 5. Results on synthetic data are reported in section 6.

2. ICA OPTIMIZATION PROBLEM FOR MOVING SOURCES

In this section we propose an optimization method to handle ICA when the unmixing matrices are time-variant. The approximation we use to model this time-variant change is a piecewise linear approximation. In the next paragraphs we will depart from the original optimization problem of time-invariant ICA and change it to a time-variant ICA optimization problem by handling equality constraints.

The original ICA optimization problem for time-invariant unmixing matrices is ([2]):

$$\max_{W} L(W) \tag{1}$$

where

$$L(W) = \frac{1}{R} \sum_{r=0}^{R-1} f_r(W) \tag{2}$$

$$f_r(W) = log\left(|W|\, g_s(W x_r)\right) \tag{3}$$

where $L(W)$ is the objective function, W is the optimization variable, r is the time index, R is the total number of samples, x_r is the vector of observation mixtures, s is the vector of sources and g_s is the probability density function of the vector of sources s.

According to [2], the batch gradient ascent update rule for this update problem is determined by the following algorithm:

$$v_r = \begin{bmatrix} \frac{g'_{s_1}(s_{1r})}{g_{s_1}(s_{1r})} \\ \cdots \\ \frac{g'_{s_N}(s_{Nr})}{g_{s_N}(s_{Nr})} \end{bmatrix} \tag{4}$$

$$\nabla_W f_r(W) = W^{-T} + v_r x_r^T \tag{5}$$

$$\nabla_W L(W) = \frac{1}{R} \sum_{r=0}^{R-1} \nabla_W f_r(W) \tag{6}$$

$$W := W + \mu \nabla_W L(W) \tag{7}$$

where N is the number of dimensions of x, s and v; and g_{s_i} is the pdf of the scalar variable s_{ir} at any time r.

2.1. Piecewise linear Approximation

While we want the un-mixing matrix to be able to change in time, we don't want to have to optimize a different un-mixing matrix at each time interval, since the number of variables in the optimization problem would be too high. We also don't want to divide the mixtures into small frames and run a different optimization problem for each of them, since we want a robust method that takes in account all the samples in the mixtures to train the mixing matrices.

For these reasons, we propose a piecewise linear approximation of the change in the un-mixing matrices. Mathematically, this means that the unmixing matrix is now going to have a time index, and it's going to vary in the following way:

$$W_r = \begin{cases} (1 - \frac{r}{R_a - 1})W_a + \frac{r}{R_a - 1}W_b & \text{, if } 0 \le r \le R_a - 1 \\ (1 - \frac{r - R_a}{R_b - R_a - 1})W_b + \frac{r - R_a}{R_b - R_a - 1}W_c & \text{, if } R_a \le r \le R_b - 1 \\ \cdots \\ (1 - \frac{r - R_x}{R_y - R_x - 1})W_y + \frac{r - R_x}{R_y - R_x - 1}W_z & \text{, if } R_x \le r \le R_y - 1 \end{cases} \tag{8}$$

where W_a, W_b ... W_z are the new unmixing matrices that the new optimization problem will have to find. R_a, R_b, R_c ... R_y are the endpoint times of each linear piece in the approximation. For, example, from time 0 to time R_a, the unmixing matrix changes linearly in time from W_a to W_b.

We will see in the next subsection that the new algorithm doesn't operate frame by frame independently, instead, it updates parameters taking in account all the samples in the mixtures.

2.2. New Optimization Problem

In this subsection we consider an approximation of only one linear piece. The multi-piece case can be easily derived from the algorithms presented in this subsection. In this case, the optimization problem would be:

$$\max_{W_a, W_b} L_2(W_a, W_b) \tag{9}$$

where

$$L_2(W_a, W_b) = \frac{1}{R_a} \sum_{r=0}^{R_a - 1} f_r(W_r) \tag{10}$$

$$f_r(W_r) = log(|W_r| g_s(W_r x_r)) \tag{11}$$

$$W_r = (1 - \frac{r}{R_a - 1})W_a + \frac{r}{R_a - 1}W_b \tag{12}$$

From the way the optimization problem is defined, we can see that all optimization variables depend on each other and will be updated given all samples of the mixtures. We see that, even though each unmixing matrix W_a, W_b... W_z will be updated only by the vicinity pieces in the gradient ascent rule (see subsection 2.3), these updates will modify the found sources for the next iteration, and this modification will affect all the unmixing matrices in the problem.

2.3. New Gradient Ascent Rule

A strict definition of the optimization problem for our time-variant case would be:

$$\max_{W_a, W_b, W_r} \frac{1}{R_a} \sum_{r=0}^{R_a - 1} f_r(W_r) \tag{13}$$

$$W_r = (1 - \frac{r}{R_a - 1})W_a + \frac{r}{R_a - 1}W_b \tag{14}$$

We see that the optimization problem is very similar to the optimization problem of the invariant case. The main difference is the presence of equality constraints. We handle those equality constraints departing from the gradient ascent rule of the time-invariant case (formulas 7, 6 and 5) and applying the chain rule as shown in the following algorithm:

$$\nabla_{W_r} f_r(W_r) = W_r^{-T} + v_r x_r^T \tag{15}$$

$$\nabla_{W_a} f_r = (1 - \frac{r}{R_a - 1}) \nabla_{W_r} f_r(W_r) \tag{16}$$

$$\nabla_{W_b} f_r = \frac{r}{R_a - 1} \nabla_{W_r} f_r(W_r) \tag{17}$$

$$\nabla_{W_a} L_2(W_a, W_b) = \frac{1}{R_a} \sum_{r=0}^{R_a - 1} \nabla_{W_a} f_r \tag{18}$$

$$\nabla_{W_b} L_2(W_a, W_b) = \frac{1}{R_a} \sum_{r=0}^{R_a - 1} \nabla_{W_b} f_r \tag{19}$$

$$W_a := W_a + \mu \nabla_{W_a} L_2(W_a, W_b) \tag{20}$$

$$W_b := W_b + \mu \nabla_{W_b} L_2(W_a, W_b) \tag{21}$$

$$W_r := (1 - \frac{r}{R_a - 1})W_a + \frac{r}{R_a - 1}W_b \tag{22}$$

With this update rule we are able to find the closest piecewise linear approximation to the variation of the unmixing matrices. In sections 3 and 4 we will explore two additional steps to find a good initial point for the gradient search of this section.

3. APPROXIMATE WHITENING

There is still the question of if the gradient ascent algorithm of section 2 would be efficient enough to find the solution without spending too many iterations. It is described in [1] and [2] that whitening the data before starting the optimization process reduces considerably the number of iterations needed since the algorithm only has to find a rotation matrix.

In the time-variant case, we would like to find a changing but continuous whitening matrix over time, and a changing but continuous rotation matrix over time.

In this section we describe an approximate whitening process that let us calculate a continuous rotation process in time for section 4. This is the only process in the final algorithm that actually processes frames independently from each other. Furthermore, this process is designed so that the algorithm in section 4 can use a piecewise linear approximation too, instead of having to analyze separate frames of the mixtures.

Given the covariance matrix C of the mixtures, a common way to whiten the data is by multiplying it by the eigenvectors matrix of C and the square root of the inverse of the diagonal matrix containing the eigenvalues of C:

$$C = E(xx^T) \tag{23}$$

$$C = V\Lambda V^T \tag{24}$$

$$y = \Lambda^{-\frac{1}{2}} V^T x \tag{25}$$

We can see that this whitening method consists of a rotation and a rescaling of the axes. Since the rotation can be different and discontinuous for different periods of time, and since in the next section we need a continuous variation of the rotation, we use a whitening method that rotates the sample back to it's original angle, as shown in the following formula:

$$y = V\Lambda^{-\frac{1}{2}} V^T x \tag{26}$$

We can see that in this case, we are applying a rotation on the sample x, a rescaling and a rotation back (done by V) to the angle it had before. This will allow us to have a continuous rotation system to analyze in section 4.

4. LIE GROUPS AND PIECEWISE LINEAR ROTATIONS

It has been found in [3] that the use of the natural gradient considerably reduces the number of iterations necessary for convergence of ICA algorithms. Self-stabilized algorithms such as in [9] can also considerably increase the convergence rate of ICA algorithms.

What these algorithms aim to do is to modify the gradient calculated in formula 5 such that the new gradient is somehow a rotation, or an equivalent of a rotation. This means, forcing the gradient to reach the unmixing matrix to have the orthogonality characteristic $W^T W = I$.

The work presented in [10] went beyond, and, instead of restricting changes in the gradient, it elegantly defined the ICA optimization problem as a problem with an orthogonality constraint $W^T W = I$. This is an equality constraint too, and handling it implies working in the lie group space of orthogonal matrices.

The gradient ascent rule in the lie group domain is easy to calculate given the gradient ascent rule derived in section 2 and is stated by the following algorithm:

$$\nabla_W f_r(W) = W^{-T} + v_r x_r^T \tag{27}$$

$$\nabla_\Theta f_r = (\nabla_W f_r(W)) W^T - W (\nabla_W f_r(W))^T \tag{28}$$

$$\nabla_\Theta L = \frac{1}{R} \sum_{r=0}^{R-1} \nabla_\Theta f_r \tag{29}$$

$$W := e^{(\mu \nabla_\Theta L)} W \tag{30}$$

And the gradient ascent rule for the piecewise linear time-variant case would be estimated by the following algorithm:

$$\nabla_{W_r} f_r(W_r) = W_r^{-T} + v_r x_r^T \tag{31}$$

$$\nabla_{\Theta_r} f_r = (\nabla_{W_r} f_r(W_r)) W_r^T - W_r (\nabla_{W_r} f_r(W_r))^T \tag{32}$$

$$\nabla_{\Theta_a} f_r = (1 - \frac{r}{R_a - 1}) \nabla_{\Theta_r} f_r \tag{33}$$

$$\nabla_{\Theta_b} f_r = \frac{r}{R_a - 1} \nabla_{\Theta_r} f_r \tag{34}$$

$$\nabla_{\Theta_a} L_2 = \frac{1}{R_a} \sum_{r=0}^{R_a-1} \nabla_{\Theta_a} f_r \tag{35}$$

$$\nabla_{\Theta_b} L_2 = \frac{1}{R_a} \sum_{r=0}^{R_a-1} \nabla_{\Theta_b} f_r \tag{36}$$

$$\Theta_a := \Theta_a + \mu \nabla_{\Theta_a} L_2 \tag{37}$$

$$\Theta_b := \Theta_b + \mu \nabla_{\Theta_b} L_2 \tag{38}$$

$$\Theta_r := (1 - \frac{r}{R_a - 1}) \Theta_a + \frac{r}{R_a - 1} \Theta_b \tag{39}$$

$$W_r := e^{\Theta_r} \tag{40}$$

Note that the linear pieces are not defined in the unmixing matrices (W) domain but in the lie group (Θ) domain. Note also that Θ is a matrix with angles as elements and it's form is:

$$\Theta = \begin{bmatrix} 0 & \theta \\ -\theta & 0 \end{bmatrix} \tag{41}$$

5. FOUR STEPS ALGORITHM

Although mathematically correct, the gradient solution of section 2 doesn't have any guarantee to be an efficient solution. We explored a special whitening step in section 3 and a continuous rotation step in 4. Both of them are dependent on each other. Since the whitening step is done in a framewise manner, we cannot expect accurate results from the algorithm developed in 4. However, we would expect these two algorithms to give an approximate good value to use as an initial point for search for the final algorithm designed in section 2. From that reasoning, the algorithm we use is made out of the following 4 steps:

I Whiten data as described in section 3

II Find piecewise linear rotation matrices as described in section 4.

III Do a piecewise linear regression on the resulting unmixing matrices from step I and step II.

IV Perform the algorithm of section 2 using as initial matrices the starting matrices an ending matrices of each piece resulting from step III. Optimize until convergence.

6. RESULTS

In this section we present results for some of the steps of the algorithm of section 5. In this work all results were done with synthetic white data, generated using a sigmoid probability density function.

In figure 1 we show the linear variation along with the results of parts II and part III of the 4 steps algorithm. The dotted line are the unmixing matrices given by part II of the algorithm. The unmixing matrices resulting from part III are shown in solid lines, and the real unmixing matrices are shown in dashed lines. We

Table 1. Mean Square Error (MSE) results for 4-steps algorithm, time-invariant ICA (T.I. ICA), output of part III of the algorithm and framewise ICA

Algorithm	4-Steps	T.I. ICA	part III	Framewise ICA
MSE	0.0044	0.4972	0.0058	0.0831

can see that the result from part II of the algorithm can be approximated to a line, close to the real mixing system. This gives a robust enough starting points for part IV of the algorithm.

In figure 2 we show the optimization space of part III of the algorithm. There are local maxima and global maxima. Several global maxima are spaced by $\frac{\Pi}{2}$ radians as expected. We avoided getting stuck in local maxima by downsampling the optimization space before the gradient ascent algorithm.

Table 1 shows mean square error (MSE) of our 4-steps algorithm compared to a time-invariant ICA algorithm [2]. We also show the MSE of the output of part III of our algorithm, as way to show the need for part IV. Finally, we show results of a framewise ICA algorithm without post-processing. The mean square error is calculated between the original signals and the separated signal by the each of the algorithms.

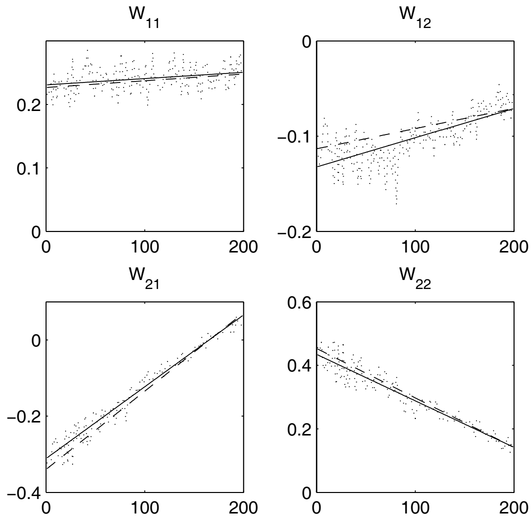

Fig. 1. Variation of unmixing Matrix W in time. Dotted: Estimated frame by frame. Solid: Linear regression approximation of frame by frame calculation. Dashed: Real variation.

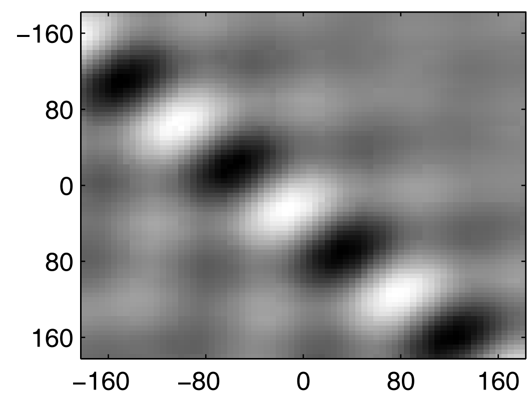

Fig. 2. Optimization space of θ_a, and θ_b.

7. CONCLUSIONS

We have shown a new approach to deal with time-variant mixing systems for blind source separation. We transformed the ICA algorithm using optimization theory to fit a piecewise linear approximation of the time-variant mixing matrices. To achieve a good choice on the initial step on the gradient descent algorithm, we performed a modified pre-whitening approach, together with a piecewise linear approximation in the lie-group domain. Results shown on synthetic data prove the superiority of this algorithm compared to time-invariant ICA algorithms. Better performance than framewise algorithms might be implied from comparing with the results reported on a simple framewise algorithm in table 1.

8. REFERENCES

[1] Aapo Hyvarinen, Juha Karhunen, Erkki Oja, "Independent Component Analysis", Wiley-Interscience, 1st Edition, 2001.

[2] A. Bell and T. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," Neural Comput. vol. 7, no. 6, pp. 1129-1159,1995.

[3] S. Amari, S. C. Douglas, A. Cichocki, H. H. Yang, "Multichannel blind deconvolution and equalization using the natural gradient," in Proc. IEEE Worksh. Signal Processing Advances in Wireless Comm., Apr 1997, pp. 101-104.

[4] P. Smaragdis, "Blind separation of convolved mixtures in the frequency domain," Neurocomput., vol. 22, pp 21-34, 1998.

[5] R.H. Lambert and A.J. Bell, "Blind separation of multiple speakers in a multipath environment," in Proc. ICASSP, Apr. 1997, pp 423-426.

[6] R. Mukai, H. Sawada, S. Araki, S. Makino, "Real-time blind source separation for moving speakers using blockwise ICA and residual crosswalk subtraction," In Proc. ICA 2003, Apr. 2003, Nara, Japan.

[7] K. E. Hild II, D. Erdogmus, J. C. Principe, "Blind source separation of time-varying, instantaneous mixtures using an on-line algorithm," In Proc. of ICASSP'02, 2002, pp. 993-996.

[8] A. Koutras, E. Dermatas and G. Kokkinakis, "Blind speech separation of moving speakers in real reverberant environments," in Proc. of ICASSP'00, 2000, pp. 1133-1136.

[9] S.C. Douglas, "Self-stabilized gradient algorithms for blind source separation with orthogonality constraints," IEEE Trans. on Neural Neworks 11(2000) 1490-1497.

[10] Mark D. Plumbley, "Lie Group Methods for Optimization with Orthogonality Constraints," Proc. ICA 2004, Granada, Spain, Sept. 22-24,2004.