Application of Augmented Lagrangian Method in Independent Component Analysis

Yiou Li*

Dec 9 2003

Abstract. In this project, a new Augmented Lagrangian method optimization algorithm is developed. And this new method is used to perform the nonlinear constrained optimization task brought up by the research of Independent Component Analysis(ICA). Experiment on the simulated data was performed, and the independent components are successfully estimated by the optimal decomposition vectors resulting from the new algorithm.

Key words. Independent Component Analysis(ICA), Augmented Lagrangian Method, LANCELOT, Line-search Newton-CG Method.

1 Introduction

1.1 Independent Component Analysis

Independent Component Analysis(ICA) is a newly developed data driven signal estimation technique. It models the observed data as a linear mixture of a finite number of statistically independent sources, and estimate those underlying sources based on two assumptions: linear mixing and statistical independence of the sources. Various algorithms have been developed by different groups, among which two streams are proved to be most successful. One is the FastICA algorithm by Hyvarinen[1], the other is the Infomax algorithm by Bell and Sejnowsky[2]. In this project, we will focus on the optimization issue of FastICA. In FastICA, the independence of a source is measured by certain nonlinear transform of the sample data from that source. By maximize/minimize the nonlinear objective function with respect to the decomposition vector, $\mathbf{w}$, the independent component can be extracted from the mixture. ICA model can be summarized by the following equations where 'S' stands for sources, 'A' is the mixting matrix and 'X' represents resulting mixtures:

$$X = A \cdot S \tag{1.1}$$

$$S = [s_1, s_2, ...s_M]^T; \tag{1.2}$$

$$X = [x_1, x_2, ...x_M]^T; \tag{1.3}$$

$$A = \begin{bmatrix} a_{11} & a_{12} & .. & a_{1M} \\ a_{21} & a_{22} & .. & a_{2M} \\ & & & \\ a_{M1} & a_{MM} & .. & a_{MM} \end{bmatrix} \tag{1.4}$$

*Dept. of Computer Science and Electrical Engineering, UMBC.

The objective function in FastICA is of the form:

$$E\{G(\mathbf{w}^T X)\} \tag{1.5}$$

where G(.) is certain nonlinear transform. In this project, we choose the nonlieарity to be:

$$G(u) = u^4 \tag{1.6}$$

This nonlinear form is chosen because: 1.It resembles the 4th order statistics which is a standard measure of how far the distribution of a random variable is from that of a Gaussian(Normal) distribution, since Gaussian distribution represents the trend of mixture of large quantity of random variables, the distribution away from Gaussian is the direction to get the independent components[1]; 2.In the context of ICA, the objective function taking this nonlinear form is convex and twice differentiable with respect to the unknowns. This is a nice property for the justification of global minimizer when implementing the optimization.

In the ICA process, a decomposition vector, $\mathbf{w}$, is to be found so the estimated sources can be calculated by the decomposition model as blow:

$$\mathbf{s}_i = \mathbf{w}_i^T X \; where \; i = 1, 2...M \tag{1.7}$$

When $\mathbf{w}$ is taken to be the unknown variable in this problem, it is easy to see that the value of the objective function goes to infinity as the value of $\mathbf{w}$ grows to be unbounded. Hence, a 'unit variance' constraint must be imposed to the optimization procedure:

$$\|\mathbf{w}\|^2 = 1 \tag{1.8}$$

Therefore, the following nonlinear constrained optimization problem for ICA is formed:

$$\mathbf{w} = arg\, min\, E\{(\mathbf{w}^T X)^4\} \; s.t. \; \|\mathbf{w}\|^2 = 1 \tag{1.9}$$

Where $E\{.\}$ takes the expectation of the random variable inside and it's value is calculated as the mean value of the data samples.

1.2 Estimate several independent components

In practical scenario, several independent components need to be estimated, and the optimization of corresponding decomposition vectors is usually carried out simultaneously. To avoid $\mathbf{w}_i$'s from converging to the same extrema of the ICA objective function, orthogonal constraints are imposed among those vectors. Putting together the 'unit variance' constraint mentioned in section 1.1, we can state all the constraints in the following matrix form:

$$WW^T = I, \tag{1.10}$$

where I denotes a M by M identity matrix.

We then take the summation of all the objective functions to be a new objective function, $F(\mathbf{w}_1, \mathbf{w}_2...\mathbf{w}_M)$, as the measure of 'total independence'. So the overall optimization issue is formed as:

$$W = arg\, min\, \Sigma E\{(W^T X)^4\} \; s.t. \; WW^T = I \tag{1.11}$$

2 Methodology

2.1 Augmented Lagrangian Method

There exits an important class of methods to solve the general constrained optimization problems like the one stated in the previous section[3]. This class of methods seeks the solution by replacing the original constrained problem with a sequence of unconstrained subproblems in which the objective function is formed by the original objective of the constrained optimization plus additional 'penalty' terms. The 'penalty' terms are made up of constraint functions multiplied by a positive coefficient. By making this coefficient larger and larger along the optimization of the sequential unconstrained subproblems, we force the minimizer of the objective function closer and closer to the feasible region of the original constrained problem[3].

However, as the penalty coefficient grows to be too large, the objective function of the unconstrained optimization subproblem may become ill conditioned, thus, making the optimization of the subproblem difficult. This issue is avoided, after the proof of convergence, by the so called 'Augmented Lagrangian method' in which an explicit estimate of the Lagrange multipliers λ is included in the objective. Hence, the objective function becomes[3]:

$$\Phi(x, \lambda, \mu) = f(x) - \Sigma\lambda_i c_i(x) + \frac{1}{2\mu}\Sigma {c_i}^2(x) \tag{2.1}$$

2.2 Modified LANCELOT Algorithm

Based on the idea of Augmented Lagrangian method, an optimization software, LANCELOT has been developed by Conn[4]. Unfortunately, this software is written in Fortran Language and has its rigid data input/output format, which makes it difficult to import the large amount of data samples from ICA problem into that software to perform optimization.

As the way out, a new Matlab code is developed based on the framework of LANCELOT. And for higher optimization efficiency, Line-search Newton-CG Method is used as the algorithm to solve the unconstrained subproblem. A framework of the modified LANCELOT algorithm is stated as below:

Algorithm 3.1(Modified LANCELOT Algo.)
Choose stopping tolerances: ϵ_i, ϵ_c;
Choose positive constants: $\eta, \omega, \mu \leq 1, \tau < 1, \gamma < 1, \alpha_w, \beta_w, \alpha_\eta, \beta_\eta, \alpha_*, \beta_*$;
satisfying $\alpha_\eta < min(1, \alpha_w), \beta_\eta < min(1, \beta_w)$;
Choose $\lambda^0 \in R^m$;
Set $\mu_0 = mu, \alpha_0 = min(\mu_0, \gamma), \omega_0 = \omega(\alpha_0)^{\alpha_w}, \eta_0 = \eta(\alpha_0)^{\alpha_\eta}$
While $\|L(x, \lambda)\| > \epsilon_i \; and \; \|c(x)\| > \epsilon_c$
Find an approximate solution x_k of the unconstrained subproblem(Augmented Lagrangian Function) by Line-search Newton-CG method such that

$$\|\nabla\Phi(x, \lambda; \mu)\| < \omega_k;$$

if $\|c(x_k)\| \leq \eta_k$
if $\|c(x_k)\| \leq \eta_* \; and \; \|\nabla L_A(x_k, \lambda_k; \mu_k)\| \leq \omega_*$
STOP with approximate solution x_k;
end (if)

(* update multipliers, tighter tolerances*)

$\lambda^{k+1} = \lambda^k - c(x_k)/\mu_k;$

$\mu_{k+1} = \mu_k;$

$\alpha_{k+1} = \mu_{k+1};$

$\eta_{k+1} = \eta_k \alpha_{k+1}^{\beta_\eta};$

$\omega_{k+1} = \omega_k \alpha_{k+1}^{\beta_\omega};$

else

(* decrease penalty parameter, tighter tolerances*)

$\lambda^{k+1} = \lambda^k;$

$\mu_{k+1} = \tau\mu_k;$

$\alpha_{k+1} = \mu_{k+1}\gamma;$

$\eta_{k+1} = \eta\alpha_{k+1}^{\beta_\eta};$

$\omega_{k+1} = \omega\alpha_{k+1}^{\beta_\omega};$

end(if)

end (while)

3 Simulation Results

3.1 Simulated fMRI data

An active application of ICA is in fMRI(functional Magnetic Resonance Imaging) analysis. In acqusition of fMRI data, the brain region is scanned while the test subjective is performing certain functional task, e.g. responsing to visual stimulus. The data is considered to be a mixture of brain hymodanymic activations from different areas in the cerebral. The ICA analysis is then used to identify those independent activation areas from the highly mixed fMRI data.

In the simulated ICA experiment, we created different image patterns to mimic the 'activation' area in human brain as well as some 'noise like' image to resemble noise with certain random distributions. Each image is considered as a random variable and the intensity at each pixel is taken as a sample data from that random variable. Since the images are created in different ways and resemble different patterns, the intensity distribution of each image are statistically independent. For the second step, we create a random mixing matrix to mix the source images. For convenience, the mixing matrix is square, so that the number of mixtures is the same as the number of independent sources. In practical cases, since the observations are highly correlated, the number of sources to be estimated is usually far less than the number of observations. For this reason, certain dimension reduction technique, e.g. Principle Component Analysis, is employed to reduce the number of observations so that ICA can be carried out in a subspace that bears most of the information from the original data[5].

3.2 Simulated fMRI data

Figure 1 illustrates the entire ICA data processing flow: The original image sources(S) are displayed in the leftmost column; the middle column shows the random mixtures(X); and the rightmost

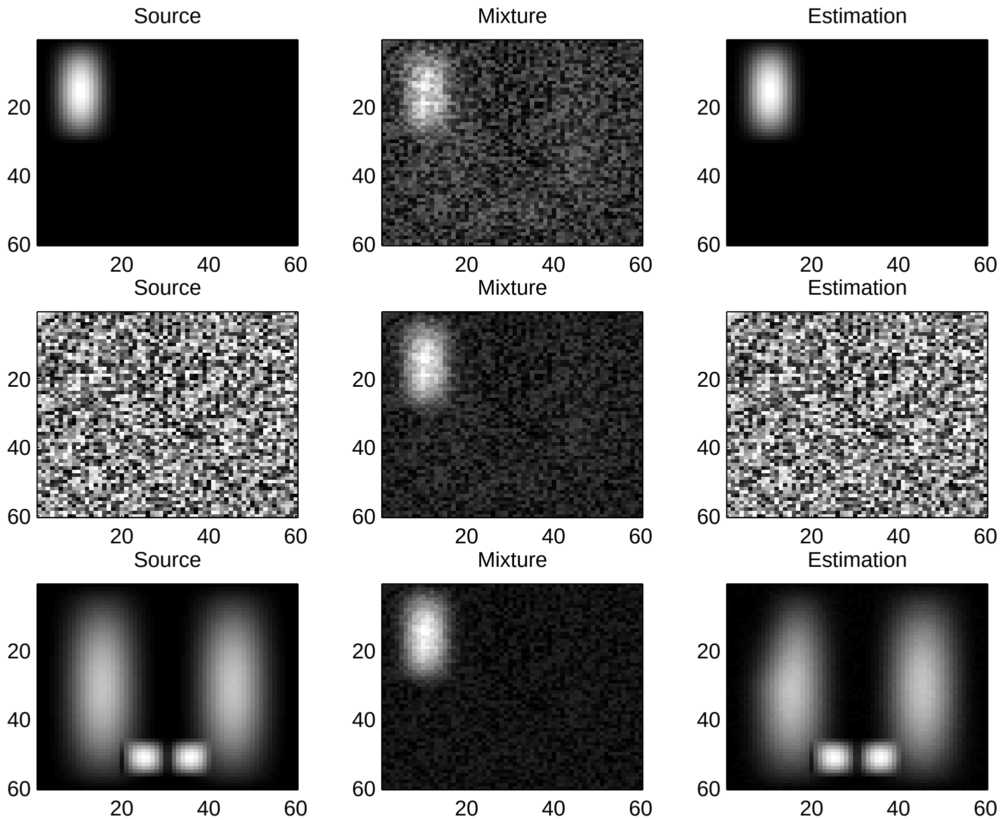

Figure 1: Original source image v.s. estimated images.

column are the estimations of the sources. The 2-D correlation between the estimated image and the corresponding source image is calculated as the benchmark for the evaluation of the estimation result. For a sample screen output of the optimization process, please refer to Appendix A.

4 Conclusion

In this project, a new nonlinear constrained optimization algorithm in the context of ICA was developed. Comparing to the classical algorithms for optimization of ICA objective functions, this new algorithm incorporates the orthogonal constraint of the decomposition vectors into the general optimization framework and perform an ensemble optimization to get several independent components by one optimization process. Being applied to the simulated fMRI image data, the algorithm produces decent result. After code improvement and testing with extensive data sources, it can be introduced as a new optimization approach in ICA research.

5 Future Work

For a certain independent measure(e.g. 4th oder statistics), the underlying components may resemble different scales or even different sign in their measured score. This difference can mislead the optimization algorithm from converging to the right independent sources. Therefore, a scaling vector could be added into the objective function according to the prior knowledge of the the underlying sources. However, proper method needs to be developed on how to decide the value of the scaling vectors through the optimization process. This leads to part of the future work on this ICA optimization algorithm.

Acknowledgments

This project is a try out of the skills learned from Course MATH651. Through which, I learned a lot information and techniques in the field of numerical optimization. This is very relevant and useful for my research work. Thanks Dr. Nayakkankuppam for the advices on my homeworks and final project. Also appreciate Dr. Gobbert and Dr. Rouben for their online instructions on how to use LaTeX. They provide handy references for my technical writing practice.

A Sample Screen Output Of The Optimization Algorithm

```
Line-search Newton-CG for subproblem:
-----------------------------------------------------
Step          f(xk)           ||d_Phy||          ||grad_L||

[50]:         -3.1362e+000     1.3350e-001,     1.5576e+001

[100]:         -3.1723e+000     7.3180e-002,     1.5559e+001

[150]:         -3.1824e+000     3.6533e-002,     1.5552e+001

[200]:         -3.1848e+000     1.7926e-002,     1.5548e+001

[250]:         -3.1853e+000     8.7499e-003,     1.5548e+001

Summary of the outer loop:
-------------------------------------------------------------------------
Step:     f(X)        sum(lambda)     mu           ||C(X)||     ||grad L||
  #1;    -5.1290e+000   6.0000e-001   1.0000e-001  6.1395e-001  4.7709e+000



Line-search Newton-CG for subproblem:
-----------------------------------------------------
Step          f(xk)           ||d_Phy||          ||grad_L||

[50]:         -2.0941e+000     3.2159e-002,     8.0600e+000

[100]:         -2.0941e+000     4.5836e-003,     8.0598e+000

[150]:         -2.0941e+000     2.6733e-003,     8.0598e+000
```

```
[200]:        -2.0941e+000     2.3542e-003,     8.0598e+000

[250]:        -2.0941e+000     2.1690e-003,     8.0598e+000


Summary of the outer loop:
-------------------------------------------------------------------------
Step:     f(X)        sum(lambda)     mu          ||C(X)||     ||grad L||
  #2;    -2.1761e+000   6.0000e-001   1.0000e-002  3.9567e-002  1.5548e+001



Line-search Newton-CG for subproblem:
-----------------------------------------------------
Step         f(xk)           ||d_Phy||          ||grad_L||

[50]:        -2.0189e+000     1.1063e-002,     3.1259e-001

[100]:        -2.0189e+000     2.3307e-003,     3.1263e-001

[150]:        -2.0189e+000     1.3675e-003,     3.1263e-001

[200]:        -2.0189e+000     1.1397e-003,     3.1263e-001

[250]:        -2.0189e+000     1.0117e-003,     3.1263e-001


Summary of the outer loop:
-------------------------------------------------------------------------
Step:     f(X)        sum(lambda)     mu          ||C(X)||     ||grad L||
  #3;    -2.0128e+000   4.2563e+000   1.0000e-002  1.5645e-003  2.0110e-003



Line-search Newton-CG for subproblem:
-----------------------------------------------------
Step         f(xk)           ||d_Phy||          ||grad_L||

[50]:        -2.0188e+000     4.7821e-003,     3.0197e-001

[100]:        -2.0188e+000     3.9491e-003,     3.0197e-001

[150]:        -2.0188e+000     3.2135e-003,     3.0197e-001

[200]:        -2.0188e+000     2.6171e-003,     3.0197e-001

[250]:        -2.0188e+000     2.1393e-003,     3.0197e-001
```

```
Summary of the outer loop:
--------------------------------------------------------------------------
Step:     f(X)        sum(lambda)     mu          ||C(X)||     ||grad L||
  #4;    -2.0182e+000   4.2563e+000   1.0000e-003  1.5100e-004  3.1263e-001



 Kurtosis of S1: 16.25;   Kurtosis of S_est1: 16.26;

 Correlation of No.1 source estimated: 1.00(+/-0.00)
.....................

 Kurtosis of S2: -1.21;   Kurtosis of S_est2: -1.21;

 Correlation of No.2 source estimated: 1.00(+/-0.00)
.....................

 Kurtosis of S3: -0.37;   Kurtosis of S_est3: -0.29;

 Correlation of No.3 source estimated: 0.99(+/-0.00)
```

References

[1] A. Hyvarinen, E. Oja 'Independent Component Analysis: algorithms and applications', Neural Networks(2000) 411-430;

[2] A.J. Bell and T.J. Sejnowski 'An Information-maximization approach to blind separation and blind deconvolution',1995;

[3] Nocedal and Wright 'Numerical Optimization', Chapter17, 1999;

[4] Conn, Gould and Toint 'LANCELOT A Fortran Package for Large-Scale nonlinear Optimization(Release A)',1992;

[5] V. D. Calhoun 'Independent Component Analysis for Functional Magnetic Resonance Imaging' Doctoral thesis, 2002;

[6] Lamport 'A Document Preparation System LaTeX User's Guide And Reference Manual(Second Edition)';