LEARNING IN MANIFOLDS: THE CASE OF SOURCE SEPARATION

Jean-François Cardoso

CNRS / ENST TSI
46 rue Barrault, 75634 Paris, France.
cardoso@sig.enst.fr
http://sig.enst.fr/~cardoso/stuff.html

ABSTRACT

The blind signal separation (BSS) problem has a distinctive feature: the unknown parameter being an invertible matrix, the parameter set is a multiplicative group and the observations can be modeled by a *transformation model*. For this reason, it is possible to design on-line algorithms which are very simple and still offer excellent performance (typically: Newton-like performance at a gradient-like cost). This paper presents two apparently different approaches to deriving these algorithms from the maximum likelihood principle. One approach (relative gradient) starts with focus on the group structure and eventually introduces the statistical structure. The other approach (natural gradient) applies to any statistical manifold and is eventually made tractable by exploiting the group structure. The relationship between these approaches is explained.

1. BACKGROUND.

1.1. Source separation.

Blind source separation (BSS) or independent component analysis (ICA) aims at computing a linear decomposition of a random vector x into components which are 'as independent as possible'. The simplest underlying model is

$$x = As \tag{1}$$

where s is an unobserved $n \times 1$ vector of independent components and $A \in \mathrm{GL}(n)$ which denotes the general linear group on $\mathbb{R}^n$ *i.e.* $\mathrm{GL}(n)$ is the set of $n \times n$ invertible matrices.

1.2. Maximum likelihood.

Assume that the probability distribution of the source vector s has a density $r(s)$ with respect to Lebesgue measure on $\mathbb{R}^n$; then in model (1), the log-density of x is

$$\log p(x; A) = \log r(A^{-1}x) - \log |A| \tag{2}$$

where $|A|$ denotes the absolute value of $\det(A)$. The derivative of the log-density is found to be

$$\frac{\partial \log p(x; A)}{\partial A} = -A^{-\dagger} H(A^{-1}x)$$

where † denotes transposition and $H : \mathbb{R}^n \mapsto \mathbb{R}^{n \times n}$ is the vector-to-matrix mapping:

$$H(y) \stackrel{\text{def}}{=} \psi(y)y^\dagger - I_n \tag{3}$$

with $\psi : \mathbb{R}^n \mapsto \mathbb{R}^n$ the vector-to-vector mapping:

$$\psi(y) \stackrel{\text{def}}{=} -\frac{\partial \log r(s)}{\partial s}.$$

If T samples $\{x(t)\}_{t=1}^T$ are available and assumed to be independently distributed, the maximum likelihood estimate (MLE) $\hat{A}_{ML}$ of A for model (1) then is solution of the *estimating equation* [1]:

$$\frac{1}{T}\sum_{t=1}^{T} H(y_t) = 0 \quad \text{with} \quad y_t = \hat{A}_{ML}^{-1} x_t. \tag{4}$$

For the sake of robustness, one may also maximize the likelihood after spatial whitening (or 'sphering'). The resulting ML estimate, under the constraint that the estimated vector y_t is spatially white, is a solution of the estimating equation (4) with $H(y)$ replaced by

$$H_o(y) \stackrel{\text{def}}{=} yy^\dagger - I_n + \varphi(y)y^\dagger - y\varphi(y)^\dagger \tag{5}$$

which is the class of functions considered in [2].

1.3. Adaptive algorithms.

We consider adaptive BSS algorithms which update an $n \times n$ 'separating matrix' B *i.e.* an estimate of the inverse of A such that the 'output' $y = Bx$ provides (after convergence) an estimate of the source vector s. A very general class of such algorithms is

$$B_{t+1} = B_t - \mu_t \, \tilde{G}(x_t, B_t)$$

Centre National de la Recherche Scientifique and École Nationale Supérieure des Télécommunications where $\tilde{G}(x, B)$ is a matrix-valued function and $\{\mu_t\}$ is a sequence of positive learning steps. We shall not concern ourselves with the sequence of learning steps but only with the design of appropriate function $\tilde{G}(x, B)$. Without loss of generality, we set $\tilde{G}(x, B) = G(Bx, B)B$ so that the learning rule is rewritten in a 'multiplicative fashion' as

$$B_{t+1} = [I_n - \mu_t G(y_t, B_t)]\, B_t$$

with $y_t = B_t x_t$.

A very important special case is when function G does *not* depend on its second parameter. Then, we simply write $G(y, B) = G(y)$ and the learning rule becomes

$$B_{t+1} = [I_n - \mu_t G(y_t)]\, B_t \qquad (6)$$

This class (6) of algorithms is uniquely determined by the mapping $G : \mathbb{R}^n \mapsto \mathbb{R}^{n \times n}$ and enjoys the desirable property of 'equivariance': its performance as an adaptive separator is independent of the particular value of the mixing matrix A and in particular it is independent of the condition number of A [2, 3, 4].

1.4. Equivariant on-line likelihood maximization.

The stationary points of the equivariant adaptive algorithm (6) are characterized by the condition

$$\mathrm{E}G(y) = 0 \quad \text{with} \quad y = Bx$$

while the stationary points (4) of the likelihood are characterized by

$$\frac{1}{T}\sum_{t=1}^{T} H(y_t) = 0 \quad \text{with} \quad y_t = \hat{A}_{ML}^{-1} x_t.$$

The similarity between these equations suggests that an appropriate G function for an adaptive algorithm is H which is obtained by differentiating the log-likelihood; however, a 'regular' stochastic gradient algorithm for maximizing the log-likelihhod does *not* result in an equivariant procedure like (6); in order to derive a 'good' function G related to ML estimation, one must use *alternate* definitions of the gradient taking into account the statistical structure of the BSS problem.

In this paper, we review the 'natural gradient' [5] and the 'relative gradient' [2]. Both approaches recast the adaptive BSS problem as a trajectory on a manifold and yield simple equivariant algorithms in the form (6). We also describe how these two approaches are related.

2. NATURAL GRADIENT

This section recalls the natural gradient approach, the resulting algorithm for adaptive BSS and how it can be reduced (at some cost) to a simple learning rule.

2.1. Gradient algorithm in a statistical manifold.

We repeat here the arguments of Amari [5]. Consider the problem of maximizing a function $f(\theta)$ with $\theta \in \mathbb{R}^p$ by a hill-climbing technique. At a given starting point θ, the direction $\delta\theta$ of steepest ascent may be found as the maximizer of $f(\theta + \delta\theta)$ for all $\|\delta\theta\| < \epsilon \ll 1$. Of course, this is the direction of the gradient $\frac{\partial f}{\partial \theta}$ and the corresponding algorithm is to change θ by adding to it the increment $\delta\theta = \mu \frac{\partial f}{\partial \theta}$ for some small $\mu > 0$:

$$\theta \leftarrow \theta + \mu \frac{\partial f}{\partial \theta}.$$

In many problems, the parameter θ is arbitrary in the sense that it results from choosing one possible parameterization of, say, a linear system. Reparameterization does not change the optimization problem but it does change in general the behavior of the gradient algorithm based on it because the constraint $\|\delta\theta\| < \epsilon$ is usually not invariant by reparameterization. In many instances however, it exists a 'meaningful' local metric

$$\|\delta\theta\|_W^2 = \delta\theta^\dagger W \delta\theta$$

with W a positive matrix possibly depending on θ. In this case, one should update θ by the increment $\delta\theta$ most increasing f among all increments of a given size: $\|\delta\theta\|_W < \epsilon$. This approach results in the gradient algorithm

$$\theta \leftarrow \theta + \mu W^{-1} \frac{\partial f}{\partial \theta},$$

which includes as a special case Newton algorithms when W is minus the Hessian of f at current point θ.

A *statistical manifold* is a smooth parametric family $p(x; \theta)$ of probability distributions seen as a differentiable manifold and equipped with the Fisher information matrix J_θ:

$$J_\theta \overset{\text{def}}{=} \mathrm{E}_\theta \frac{\partial \log p(x, \theta)}{\partial \theta} \frac{\partial \log p(x, \theta)}{\partial \theta}^\dagger$$

as the local metric. The argument of previous paragraph apply with $f(\theta) = \log p(x, \theta)$ and $W = J_\theta$. This makes a lot of sense because the effect of multiplication by J_θ^{-1} is to make the update larger in directions in which the variations of θ have less *statistical* significance. The resulting update is

$$\delta\theta = \mu \left(\mathrm{E}_\theta \frac{\partial \log p(x, \theta)}{\partial \theta} \frac{\partial \log p(x, \theta)}{\partial \theta}^\dagger \right)^{-1} \frac{\partial \log p(x, \theta)}{\partial \theta} \qquad (7)$$

and, for 'small' steps, is invariant under reparameterization. This is termed the 'natural gradient learning' [5] and is the Fisher method of scoring when $\mu = 1$.

2.2. Natural gradient in a group

Natural gradient updates, as introduced in previous section, can be considered in any parametric model (see several examples in [5]) but may be difficult to implement because they usually require the estimation and the inversion of the Fisher information matrix. However, in the particular case of BSS, the group structure comes nicely into play: calculations [5] show that the updating rule (7) applied to the BSS likelihood yields an on-line algorithm which is precisely in the form of (6) with function $G(y)$ given by

$$\overline{G}(y) = \left(\mathrm{E}_r \overline{H}(s)\overline{H}(s)^\dagger\right)^{-1} \overline{H}(y) \tag{8}$$

In eq. (8), an bar $\bar{\cdot}$ denotes the vectorization of a matrix into a column vector ($\overline{G} \overset{\text{def}}{=} \mathrm{Vec}(G)$), function H is as defined in (3) and E_r denotes expectation under the model, that is for s distributed with a density $r(s)$ as in eq. (2) for instance. In essence, what happens is that all parameter-dependent quantities are 'shifted' to the origin by action of the group (the same phenomenon occurs in optimal cumulant matching for BSS [6]). As a result, and this is a key point is that matrix $\mathrm{E}_r \overline{H}(s)\overline{H}(s)^\dagger$ is fixed: it does *not* depend on the model parameters; further, it can be explicitly inverted thanks to its block-diagonal structure (see [1, 7] and [2] when $H(\cdot) = H_o(\cdot)$ as in eq. (5)).

Another point is that $\mathrm{E}_r \overline{H}(s)\overline{H}(s)^\dagger$ being a positive matrix it can be replaced by I_{n^2} (the $n^2 \times n^2$ identity matrix) without affecting the local stability of the algorithm. This simplification reduces $G(y)$ to $G(y) = H(y)$ which is the starting point of next section.

3. RELATIVE GRADIENT

We consider first the generic problem of optimizing a function $f(\theta)$ whose parameter θ belongs to a continuous group $\mathcal{G}$. We exhibit gradient algorithms in the group and then specialize the idea to the BSS problem where the unknown parameter is an invertible matrix and $\mathcal{G} = \mathrm{GL}(n)$.

3.1. Gradient algorithm in a group.

Let $\mathcal{G}$ be a p-dimensional continuous group with composition law $\circ$ and unit element i. Denote $\tau : \mathbb{R}^p \mapsto \mathcal{G}$ a smooth mapping from a neighborhood of 0 in R^p to a neighborhood of i in $\mathcal{G}$ and such that $\tau(0) = i$. Such a map can be used to parameterize the neighborhood of the unit element i in the group. The parameterization is just $\mathcal{E} \to \tau(\mathcal{E})$ where $\tau(\mathcal{E})$ is a transformation 'close' to the identity whenever $\mathcal{E}$ is a 'small' vector of $\mathbb{R}^p$. The mapping τ can also be used to parameterize the neighborhood of *any* other element of the group by a simple 'shift' *i.e.* $\mathcal{E} \to \tau(\mathcal{E}) \circ \theta$ maps a neighborhood of 0 in $\mathbb{R}^p$ to a neighborhood of θ in $\mathcal{G}$. This defines a system of local coordinates at each point θ in $\mathcal{G}$. It can be used in particular in adaptive algorithm for updating a 'current value' of θ^n into θ^{n+1} as

$$\theta^{n+1} = \tau(\mu \mathcal{E}^n) \circ \theta^n \tag{9}$$

where μ is a a learning step and $\mathcal{E}^n$ is the direction of the update. We now examine how such a direction is obtained in a gradient algorithm.

We consider a gradient technique for the maximization of a function $f : \mathcal{G} \mapsto \mathbb{R}$ defined on the group. For a given map τ parameterizing the neighborhood of the identity, define the *relative gradient* of f at θ as the $p \times 1$ vector $\nabla_\tau f(\theta)$:

$$\nabla_\tau f(\theta) = \left.\frac{\partial f(\tau(\mathcal{E}) \circ \theta)}{\partial \mathcal{E}}\right|_{\mathcal{E}=0} \tag{10}$$

which characterizes the first order (in $\mathcal{E}$) change of $f(\theta)$ when θ is shifted by $\tau(\mathcal{E})$. A relative gradient algorithm for maximizing $f(\theta)$ by the scheme (9) is to update in the direction of the relative gradient (10): the relative gradient algorithm in the group updates θ^n into θ^{n+1} by

$$\theta^{n+1} = \tau(\mu \nabla_\tau f(\theta^n)) \circ \theta^n \tag{11}$$

with μ a positive learning step. In problems similar to BSS, this rule yields uniform performance algorithms. An abstract treatment is given in [8] (illustrated by the group of translation-scale transforms) but due to lack of space, only the BSS model is described here. The idea can also be used with the convolutive group [9] (blind equalization) but some approximations are then necessary in practice).

3.2. Relative gradient in a statistical group.

We now specialize to the BSS problem *i.e.* $\theta = A^{-1} = B$, $\mathcal{G} = \mathrm{GL}(n), i = I_n, p = n^2$. The simplest parameterization of $\mathrm{GL}(n)$ around $i = I_n$ is

$$\tau_\star : \mathbb{R}^{n \times n} \mapsto \mathrm{GL}(n), \qquad \tau_\star(\mathcal{E}) \overset{\text{def}}{=} I_n + \mathcal{E}. \tag{12}$$

The corresponding relative gradient is denoted $\nabla_\star \overset{\text{def}}{=} \nabla_{\tau_\star}$; it is the gradient defined in [2]. One finds:

$$\nabla_\star \log p(x; A) = -H(y) \quad \text{with} \quad y = A^{-1}x = Bx$$

with H defined in (3) Combining this with eqs. (12) and (11) yields exactly the equivariant algorithm (6) with $G(y) = H(y)$ which thus appears as a stochastic relative gradient algorithm for maximizing the BSS likelihood.

How does the relative gradient algorithm change when another parameterization τ is used in place of $\tau_\star$? For a given parameterization τ of the neighborhood of i, let D denote the $n^2 \times n^2$ derivative matrix

$$D = \left.\frac{\partial \overline{\tau}(\mathcal{E})}{\partial \overline{\mathcal{E}}}\right|_{\mathcal{E}=0} \tag{13}$$

where, as in eq. (8), a bar denotes the vectorization of a matrix. It is readily checked that ∇_τ is related to $\nabla_\star$ by

$$\overline{\nabla_\tau} = D^\dagger \, \overline{\nabla_\star}.$$

Using the derivative (13), the first-order Taylor expansion of the parameterization is $\overline{\tau}(\mathcal{E}) \approx \overline{I}_n + D\overline{\mathcal{E}}$. For small enough learning steps, the resulting stochastic relative gradient algorithm is again found to be in the form (6) with $G(y)$ given by

$$\overline{G}(y) = DD^\dagger \overline{H}(y). \qquad (14)$$

Matrix $DD^\dagger$ being positive, the local stability conditions are identical for $G(y) = H(y)$ and for $G(y)$ given by (14). Equivariance is also preserved because D is a constant matrix.

The statistical structure of the BSS model may be used to select a particular parameterization. For instance, one may choose $\tau(\mathcal{E})$ in such a way that the Kullback divergence $K[\tau(\mathcal{E})s\|s]$ between the distribution of s and the distribution of $\tau(\mathcal{E})s$ is equal at first order to the Euclidean norm of $\mathcal{E}$. Simple expansions show that if

$$K[\tau(\mathcal{E})s\|s] = \frac{1}{2}\|\mathcal{E}\|^2 + o(\|\mathcal{E}\|^2),$$

then the derivative matrix D should satisfy

$$DD^\dagger = \left(\mathrm{E}_r \overline{H}(s)\overline{H}(s)^\dagger\right)^{-1}.$$

This is precisely the factor which appears in the natural gradient algorithm at eq. (8).

4. DISCUSSION AND CONCLUSIONS

The BSS problem can be solved on-line by equivariant algorithms, depending only a field $G(y)$. Two apparently different ideas –relative gradient and natural gradient– lead to such algorithms (*i.e.* to a particular field $G(y)$) by considering a stochastic gradient ascent of the log-likelihood.

The natural gradient builds upon the Riemannian structure of the model (statistical manifold). It uniquely determines the field but is usually difficult to implement; the group structure of the BSS problem, however, gives a much simpler structure (8) to the field. This can be further simplified to $\overline{G}(y) = \overline{H}(y)$ *i.e.* $G(y) = H(y)$ which is the 'canonical' solution offered by relative gradient approach.

The relative gradient builds upon the Lie structure of the model (continuous group). It is determined by any parameterization of the neighborhood of the identity. The simplest parameterization yields the field $G(y) = H(y)$; a statistically significant parameterization yields the same solution (8) as the natural gradient.

5. REFERENCES

[1] D.-T. Pham and P. Garat, “Blind separation of mixture of independent sources through a quasi-maximum likelihood approach,” *IEEE Tr. SP*, vol. 45, pp. 1712–1725, July 1997.

[2] J.-F. Cardoso and B. Laheld, “Equivariant adaptive source separation,” *IEEE Trans. on Sig. Proc.*, vol. 44, pp. 3017–3030, Dec. 1996.

[3] J.-F. Cardoso, “The equivariant approach to source separation,” in *Proc. NOLTA*, pp. 55–60, 1995.

[4] A. Cichocki, R. Unbehauen, L. Moszczynski, and E. Rummert, “A new on-line adaptive learning algorithm for blind separation of source signals,” in *Proc. ISANN*, pp. 406–411, 1994.

[5] S.-I. Amari, “Natural gradient works efficiently in learning,” *Neural Computation*, vol. 10, pp. 251–276, 1998.

[6] J.-F. Cardoso, S. Bose, and B. Friedlander, “On optimal source separation based on second and fourth order cumulants,” in *Proc. IEEE Workshop on SSAP, Corfou, Greece*, 1996.

[7] S.-I. Amari, T.-P. Chen, and A. Cichocki, “Stability analysis of adaptive blind source separation,” *Neural Networks*, vol. 10, no. 8, pp. 1345–1351, 1997.

[8] J.-F. Cardoso and S.-I. Amari, “Maximum likelihood source separation: equivariance and adaptivity,” in *Proc. of SYSID'97, 11th IFAC symposium on system identification, Fukuoka, Japan*, pp. 1063–1068, 1997.

[9] S.-I. Amari, S. Douglas, A. Cichocki, and H.H.Yang, “Multichannel blind deconvolution and equalization using the natural gradient,” in *IEEE workshop on Signal Processing Advances in Wireless Communications*, pp. 101–104, Apr. 1997.