Accelerating likelihood optimization for ICA on real signals

Pierre Ablin, Jean-François Cardoso, Alexandre Gramfort

Accelerating likelihood optimization for ICA on real signals

Pierre Ablin[1], Jean-François Cardoso[2], and Alexandre Gramfort[1]

[1] Inria, Université Paris-Saclay, France
[2] Institut d'Astrophysique de Paris / CNRS, France

Abstract. We study optimization methods for solving the maximum likelihood formulation of independent component analysis (ICA). We consider both the the problem constrained to white signals and the unconstrained problem. The Hessian of the objective function is costly to compute, which renders Newton's method impractical for large data sets. Many algorithms proposed in the literature can be rewritten as quasi-Newton methods, for which the Hessian approximation is cheap to compute. These algorithms are very fast on simulated data where the linear mixture assumption really holds. However, on real signals, we observe that their rate of convergence can be severely impaired. In this paper, we investigate the origins of this behavior, and show that the recently proposed Preconditioned ICA for Real Data (Picard) algorithm overcomes this issue on both constrained and unconstrained problems.

Keywords: Independent component analysis, maximum likelihood estimation, preconditioning, optimization

1 Introduction

Linear Independent Component Analysis (ICA) [1] is an unsupervised data exploration technique, which models the set of observed signals as a linear instantaneous mixture of independent sources. Several methods have been proposed in the literature for recovering the sources and mixing matrix. When formulated as a maximum likelihood estimation task, ICA becomes an optimization problem where the negative log-likelihood has to be minimized. ICA may constitute a bottleneck in practical data processing pipelines, for example due to very long signals, high number of sources or bootstrapping techniques [2]. It is hence crucial to maximize the likelihood as quickly as possible.

Several approaches are found in the literature. Infomax [3] can be seen as a stochastic gradient descent [4]. Several second order methods have also been proposed. In [5], the author propose a quasi-Newton method dubbed "Fast Relative Newton" method, which we will refer to as "FR-Newton" in the following. In [6], a trust-region technique is used. AMICA [7] also uses a quasi-Newton approach. Although it is formulated as a fixed point algorithm, FastICA [8] is a maximum likelihood estimator under whiteness constraint of the signals [9], and also behaves like a quasi-Newton method close to convergence [10].

The aforementioned algorithms all share the following property: the Hessian approximation that they use (implicitly or explicitly) stems from the expression that the true Hessian takes when the problem is solved, *i.e.* when the signals are truly independent. Unfortunately, in most practical cases, the assumption that the observed signals are a mixture of independent signals is false to some extent. There might be fewer/more sources than observed signals, the sources might not be i.i.d. or stationary, they might be partially correlated, or there might be some convolutive mixture.

In the following, we demonstrate that this can lead to large differences between the true Hessian and its approximations, often leading to slow convergence on real data. We then show that the recently proposed Preconditioned ICA for Real Data (Picard) algorithm [11, 10] overcomes this problem and is able to build a better Hessian approximation.

This article is organized as follows. In section 2, we recall the maximum likelihood formulation of ICA, study the objective function, and derive a classical Hessian approximation. In section 3, we give some classical results about quasi-Newton algorithms, and show how the convergence speed is linked with the distance between the true Hessian and the approximation. Section 4 contains a brief description of the Picard algorithm. Finally, we illustrate the previous result with experiments in section 5. We show that Picard builds a much better Hessian approximation that those used in previous algorithms. Through extensive experiments, we show that this leads indeed to faster convergence.

Notation The mean of a time-indexed sequence $x(t)_{t=1..T}$ is noted $\hat{E}[x(t)] \triangleq \frac{1}{T}\sum_{t=1}^{T} x(t)$, and its expectation is noted $\mathbb{E}[x]$. When M is a square $N \times N$ matrix, $\exp(M)$ denotes its matrix exponential, defined as $\exp(M) \triangleq \sum_{n=0}^{\infty} \frac{M^n}{n!}$. For two $N \times N$ matrices M and M', we use the Frobenius scalar product: $\langle M|M'\rangle \triangleq \sum_{i,j} M_{ij}M'_{ij}$. We denote by $||M|| \triangleq \sqrt{\langle M|M\rangle}$ the associated norm. For a fourth order tensor H of size $N \times N \times N \times N$, the scalar product with respect to H is defined as $\langle M|H|M'\rangle \triangleq \sum_{i,j,k,l} H_{ijkl}M_{ij}M'_{kl}$ The *spectrum* $\mathrm{Sp}(B)$ of a linear symmetric operator B is the set of its eigenvalues. The Kronecker symbol δ_{ij} is equal to 1 when $i = j$ and to 0 otherwise.

2 Maximum-likelihood ICA

In this section, we derive the maximum-likelihood formulation of ICA, and study the underlying objective function.

2.1 Objective function

One observes N temporal signals $x_1(t), \cdots, x_N(t)$ of T samples each. The signal matrix is $X = [x_1(t), \cdots, x_N(t)]^\top \in \mathbb{R}^{N\times T}$.

For the rest of this article, we assume without loss of generality that X is white, *i.e.* the covariance $C \triangleq \frac{1}{T}XX^\top = I_N$. This can be enforced by a preprocessing whitening step: multiplying X by a square root inverse of C.

The linear ICA model considered here is the following [1]: there are N statistically independent and identically distributed signals, $s_1(t), \cdots, s_N(t)$, which are noted as $S \in \mathbb{R}^{N\times T}$ in matrix form, and an invertible matrix $A \in \mathbb{R}^{N\times N}$ such that $X = AS$. The s_i are referred to as sources, and A is called the mixing matrix. The aim is to estimate A and S given X. In the following, p_i denotes the probability density function (p.d.f.) of the i-th source s_i.

The likelihood of A writes [12]:

$$p(X|A) = \prod_{t=1}^{T} \frac{1}{|\det(A)|} \prod_{i=1}^{N} p_i([A^{-1}X]_{it}) \ . \tag{1}$$

It is more practical to work with the averaged negative log-likelihood, and the variable $W = A^{-1}$ called the *unmixing matrix*. In the following, $Y \triangleq WX$ denotes the current estimated sources. We define $\mathcal{L}(W) \triangleq -\frac{1}{T}\log(p(X|W^{-1}))$. It writes:

$$\mathcal{L}(W) = -\log|\det W| + \sum_{i=1}^{N} \hat{E}[-\log(p_i(Y_{it}))] \ , \tag{2}$$

where $\hat{E}$ denotes the time-averaging operation. FastICA attempts to minimize $\mathcal{L}(W)$ under whiteness constraint $WW^\top = I_N$.

2.2 Relative gradient and Hessian

To study the variations of $\mathcal{L}$, it is convenient to work in a relative framework [13], where the gradient G and Hessian H are given by the Taylor expansion of $\mathcal{L}(\exp(\mathcal{E})W)$ where $\mathcal{E}$ is a small $N \times N$ matrix. G and H are implicitly defined by the equation:

$$\mathcal{L}(\exp(\mathcal{E})W) = \mathcal{L}(W) + \langle G|\mathcal{E}\rangle + \frac{1}{2}\langle \mathcal{E}|H|\mathcal{E}\rangle + \mathcal{O}(||\mathcal{E}||^3) \ . \tag{3}$$

G is a square $N \times N$ matrix, and H is a linear operator from matrices to matrices, which can be seen as a $N \times N \times N \times N$ tensor. In the following, $\psi_i \triangleq -\frac{p_i'}{p_i}$ is referred to as the *score function*. Simple computations yield (see [10] for details):

$$G(W)_{ij} = \hat{E}[\psi_i(y_i)y_j] - \delta_{ij} \ \text{ for } \ 1 \le i,j \le N \tag{4}$$

$$H(W)_{ijkl} = \delta_{il}\delta_{jk}\hat{E}[\psi_i(y_i)y_i] + \delta_{ik}\,\hat{E}[\psi_i'(y_i)y_j y_l] \ \text{ for } \ 1 \le i,j,k,l \le N \tag{5}$$

The Hessian is sparse since it has of the order of N^3 non-zero coefficients. Still, its evaluation requires computing $O(N^3)$ sample averages $\hat{E}[\psi_i'(y_i)y_j y_l]$, making the standard Newton's method impractical for large data sets.

2.3 The Hessian approximation

If the signals $(y_1(t), \cdots, y_N(t))$ are independent, then $\mathbb{E}[\psi_i'(y_i)y_j y_l] = \delta_{jl}\mathbb{E}[\psi_i'(y_i)y_j^2]$. A natural approximation of H is then :

$$= \delta_{il}\delta_{jk}\hat{E}[\psi_i(y_i)y_i] + \delta_{ik}\delta_{jl}\,\hat{E}[\psi_i'(y_i)y_j^2] \ . \tag{6}$$

Algorithm 1: Quasi-Newton method for likelihood optimization

input : Set of white mixed signals X, boolean "whiteness constraint"
Set $W = I_N$;
Set $Y = X$;
repeat
  Compute the gradient G using (4);
  if *whiteness constraint* **then**
    Project G on the antisymmetric matrices: $G \leftarrow \frac{1}{2}(G - G^\top)$;
  end
  Compute a Hessian approximation $\hat{H}$;
  Compute the search direction $D = -\hat{H}^{-1}G$;
  if *whiteness constraint* **then**
    Project D on the antisymmetric matrices: $D \leftarrow \frac{1}{2}(D - D^\top)$;
  end
  Compute the step size $\alpha = \arg\min_\alpha \mathcal{L}(\exp(\alpha D)W)$ using line-search ;
  Set $W \leftarrow \exp(\alpha D)W$;
  Set $Y = WX$;
output: Unmixing matrix W, unmixed signals Y.

This approximation matches the true Hessian **if the number of samples goes to infinity and the** (y_i) **are independent**. If the linear ICA model holds, i.e. if there exists independent signals S and a mixing matrix A such that $X = AS$, then, for $W^* = A^{-1}$, $\tilde{H}(W^*) = H(W^*) + \mathcal{O}(\frac{1}{\sqrt{T}})$. As the number of samples is generally large, the approximation is very good in that case.

However, in a practical case, ICA is performed on real data for which the ICA model does not hold exactly. In that case, even for $W^* = \arg\min \mathcal{L}(W)$, one does not necessarily have $\mathbb{E}[\psi_i'(y_i)y_j y_l] = \delta_{jl}\mathbb{E}[\psi_i'(y_i)y_j^2]$, and $\tilde{H}(W^*)$ may be quite far from $H(W^*)$.

3 Speed of convergence of quasi-Newton methods

In the following, we consider a general relative quasi-Newton method to minimize $\mathcal{L}$, described in algorithm 1. It takes as input the set of mixed signals X, which are assumed white for simplicity, and a boolean "whiteness constraint" which determines if the algorithm works under whiteness constraint. Note that the policy to compute the approximation $\hat{H}$ is not specified: one could use $\hat{H} = \tilde{H}$, but other choices are possible. To keep the analysis simple, we assume that the line-search is perfect, i.e. that the objective function is always minimized in the search direction.

3.1 Theoretical results

Let us recall some results on the convergence speed of such method. These results mostly come from Numerical Optimization [14], chapter 3.3.

First, the following theorem shows that under mild assumptions, the sequence of unmixing matrices produced by algorithm 1 converges to a local minimum of $\mathcal{L}$.

Theorem 1. *Assume that the sequence of Hessian approximations $\hat{H}$ used in algorithm 1 is positive definite, of spectrum lower bounded by some constant $\lambda_{min} > 0$. Then, the sequence of unmixing matrices generated by the algorithm converges towards a matrix W^* such that $G(W^*) = 0$ and $H(W^*)$ is positive definite.*

This theorem is a direct consequence of Zoutendijk's result (see [14], theorem 3.2). Interestingly, it implies that the algorithm cannot converge to a saddle point (where $H(W^*)$ is not positive), but only towards local minima, as guaranteed for gradient based methods.

Quasi-Newton methods typically aim at finding a direction close to Newton's direction $-H^{-1}G$, and ideally have the same quadratic convergence rate. By Theorem 3.6 in [14], this happens if and only if at convergence, the Hessian approximation matches the true Hessian in the search direction. As we have seen before, even when the ICA model holds, the simple approximation $\tilde{H}$ only matches asymptotically the true Hessian, meaning that the above theorem never practically applies. Thus, the convergence of algorithm 1 can only be linear. The following algorithm gives the rate of convergence.

Theorem 2. *Assume that the condition of theorem 1 holds. Assume that the sequence of approximate Hessians $\hat{H}$ converges towards $\hat{H}^*$. Let λ_m (resp. λ_M) be the smallest (resp. largest) eigenvalue of $\hat{H^*}^{-\frac{1}{2}} H \hat{H^*}^{-\frac{1}{2}}$ and define the condition number:*

$$\kappa \triangleq \frac{\lambda_M}{\lambda_m} \ . \tag{7}$$

Then, for all $r < \frac{1}{\kappa}$ and n large enough, the sequence W_n of unmixing matrices produced by algorithm 1 satisfies $\mathcal{L}(W_{n+1}) - \mathcal{L}(W^*) \leq (1-r)[\mathcal{L}(W_n) - \mathcal{L}(W^*)]$.

We now give a brief sketch of proof.

Proof. For simplicity, the proof is made in a non-relative framework, where the update rule is $W_{n+1} = W_n - \alpha \hat{H}_n^{-1} \nabla \mathcal{L}(W_n)$. First, we make the useful change of variable $U_n = \hat{H^*}^{\frac{1}{2}} W_n$, and define the new objective function $L(U_n) = \mathcal{L}(\hat{H^*}^{-\frac{1}{2}} U_n)$. Simple computations show that U_n verifies $U_{n+1} = U_n - \alpha B_n \nabla L(U_n)$, where $B_n \triangleq \hat{H^*}^{\frac{1}{2}} \hat{H}_n^{-1} \hat{H^*}^{\frac{1}{2}}$. This sequence tends towards identity, meaning that the behavior of U_n is asymptotically the same as a gradient descent. One has $\nabla^2 L(U) = \hat{H^*}^{-\frac{1}{2}} [\nabla^2 \mathcal{L}(W)] \hat{H^*}^{-\frac{1}{2}}$.

Let $\varepsilon > 0$ be a small number. Since $\mathrm{Sp}(B_n) \to \{1\}$ and $\mathrm{Sp}(\nabla^2 L(U_n) \subset [\lambda_m, \lambda_M]$ as n goes to infinity, for n large enough we have that $\mathrm{Sp}(B_n) \subset [1-\varepsilon, 1+\varepsilon]$ and $\mathrm{Sp}(\nabla^2 L(U_n)) \subset [(1-\varepsilon)\lambda_m, (1+\varepsilon)\lambda_M]$. This means that the iterates U_n are in a set where L is $(1+\varepsilon)\lambda_M-$smooth and $(1-\varepsilon)\lambda_m-$strongly convex. The smoothness implies the following convexity inequality:

$$L(V) \leq L(U) + \langle \nabla L(U) | V - U \rangle + \frac{(1+\varepsilon)\lambda_M}{2} ||U - V||^2 \tag{8}$$

and the strong convexity enforces the Polyak-Lojasiewicz conditions [15]:

$$\frac{1}{2} ||\nabla f(U)||^2 \geq (1-\varepsilon)\lambda_m [L(U) - L(U^*)] \tag{9}$$

Let β be a positive scalar. For an exact line-search, we have $L(U_{n+1}) \leq L(U_n - \beta B_n \nabla L(U_n))$. Using $U = U_n$ and $V = U_n - \beta B_n \nabla L(U_n)$ in inequality (8), we obtain:

$$L(U_{n+1}) - L(U_n) \leq -\beta \langle \nabla L(U_n) | B_n \nabla L(U_n) \rangle + \beta^2 \frac{(1+\varepsilon)\lambda_M}{2} ||B_n \nabla L(U_n)||^2 \tag{10}$$

The condition on the spectrum of B_n implies $\langle \nabla L(U_n) | B_n \nabla L(U_n) \rangle \geq (1-\varepsilon)||\nabla L(U_n)||^2$ and $||B_n \nabla L(U_n)||^2 \leq (1+\varepsilon)^2 ||\nabla L(U_n)||^2$. Replacing in eq. (10) yields:

$$L(U_{n+1}) - L(U_n) \leq \left(-\beta(1-\varepsilon) + \beta^2 \frac{(1+\varepsilon)^3 \lambda_M}{2} \right) ||\nabla L(U_n)||^2 \tag{11}$$

This holds for any β, in particular for $\beta = \frac{1-\varepsilon}{(1+\varepsilon)^3 \lambda_M}$ (which minimizes the scalar factor in front of $||\nabla L(U_n)||^2$). We obtain:

$$L(U_{n+1}) - L(U_n) \leq -\frac{(1-\varepsilon)^2}{2(1+\varepsilon)^3 \lambda_M} ||\nabla L(U_n)||^2 \tag{12}$$

Using eq. (9) then gives:

$$L(U_{n+1}) - L(U_n) \leq -\frac{(1-\varepsilon)^3 \lambda_m}{(1+\varepsilon)^3 \lambda_M} [L(U_n) - L(U^*)] \tag{13}$$

Rearranging the terms, we obtain the desired result for $r = (\frac{1-\varepsilon}{1+\varepsilon})^3 \frac{1}{\kappa}$.

3.2 Link with maximum likelihood ICA

There are many ICA algorithms closely related to the minimization of $\mathcal{L}$ and similar to Algorithm 1 . For instance, Infomax is a stochastic version of algorithm 1 without whiteness constraint and with $\hat{H} = Id$. In [5], the author proposes to use $\hat{H} = \tilde{H}$ in algorithm 1, without the whiteness constraint. The algorithm is denoted as "Fast Relative Newton method", or FR-Newton for short. The same approach is used in AMICA [7]. In [10], it is shown that close to convergence, FastICA's iterations are similar to those of algorithm 1 with the whiteness constraint, and where the Hessian approximation has the same properties as $\tilde{H}$: it coincides asymptotically with H when the underlying signals (y_i) are independent, but may differ otherwise. Thus, the previous results apply for a wide range of popular ICA methods.

4 Preconditioned ICA for Real Data

Let us now introduce the Preconditioned ICA for Real Data (Picard) algorithm, which finds a better Hessian approximation than $\tilde{H}$. The algorithm is an adaptation of the L-BFGS algorithm [16]. It has a memory of size m which stores the m previous iterates W and gradients G. From these values, it recursively builds a Hessian approximation starting from $\tilde{H}$. In the following, H_P denotes that approximation. It does so in an uninformed fashion, without any prior on the local geometry. L-BFGS has been shown to perform well on a wide variety of problems. Here, we have the advantage of having $\tilde{H}$ as a good initialization for the approximate Hessian. Another asset of this method is that the Hessian approximation never has to be computed, because there is an efficient way of computing the direction $-H_P^{-1}G$. Picard can handle both constrained and unconstrained problems. For further details for the practical implementation, see [11, 10].

Python and Matlab/Octave code for Picard is available online.[3]

5 Experiments

5.1 Comparison of the condition numbers

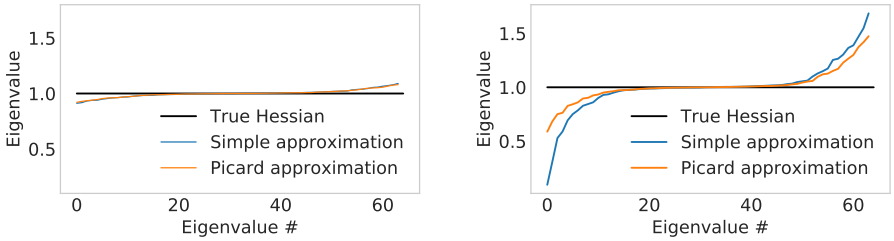

Fig. 1. A measure of the closeness of the approximate Hessians to the true Hessian at the maximum likelihood: sorted spectrum of $\hat{H}^{-\frac{1}{2}} H \hat{H}^{-\frac{1}{2}}$. Left: simulated data where the ICA model holds. Right: real data. On the simulated data, we find $\kappa = 1.2$ for both $\hat{H} = \tilde{H}$ and $\hat{H} = H_P$. For that example on real data, we find $\kappa = 29$ for $\tilde{H}$ and a significantly smaller $\kappa = 2.6$ for H_P.

In this section, we show how close the Hessian approximations $\tilde{H}$ and H_P are to H on simulated and real data. We consider two different datasets X of $N = 8$ signals of length $T = 20000$. The first one is obtained by simulating a source matrix S of independent signals, and a random mixing matrix A. We take $X = AS$. For that dataset, the linear ICA model holds by construction. The

[3] https://github.com/pierreablin/picard

second one is obtained by extracting 20000 square patches of size $(8, 8)$ from a natural image. PCA is then applied to reduce to 8 the number of signals.

First, we find a local minimum W^* of $\mathcal{L}(W)$ by running one of the algorithms on this dataset. Then, the simple approximation $\tilde{H}(W^*)$, the Picard approximation $H_P(W^*)$ and the true Hessian $H(W^*)$ are computed. As explained by theorem 2, what drives the convergence speed of the algorithms is the spectrum of $\hat{H}^{-\frac{1}{2}} H \hat{H}^{-\frac{1}{2}}$ where $\hat{H}$ is the approximation. Figure 1 displays these spectrum for the two datasets.

We observe that H_P and $\tilde{H}$ are very similar on the simulated dataset, and that the resulting condition numbers are close to 1, which explains the fast convergence of the two algorithms. On the real dataset, the results are different: the spectrum obtained with H_P is flatter than the one obtained with $\tilde{H}$, which means that Picard builds a Hessian approximation which is significantly better than $\tilde{H}$.

5.2 Convergence speed on real datasets

We now compare the convergence speed of Picard / Picard-O with FR-Newton from [5] and FastICA [9] on three types of data on which ICA is widely used.

The first is a cancer genomics dataset generated by the TCGA Research Network: `http://cancergenome.nih.gov`, of initial size $N \simeq 2000$ and $T \simeq 20000$ for which the dimension has been reduced to $N = 60$ by PCA. The second consists of 13 EEG recordings datasets [17] of size $N = 71$ and $T \simeq 300000$. The last one is 30 datasets of $T = 20000$ extracted image patches of size $(8, 8)$, flattened to obtain $N = 64$ signals. We run the aforementioned algorithms 10 times on each datasets. We keep track of the evolution of the gradient norm across iterations and time. Figure 2 displays the median and $10 - 90\%$ percentile of the trajectories.

As expected regarding the previous results on the Hessian spectrum, Picard and Picard-O converge faster than their counterparts relying purely on $\tilde{H}$ as Hessian approximation.

Conclusion

This article considers quasi-Newton methods for maximum likelihood ICA using approximated Hessian matrices. We argue that while the standard Hessian approximation works very well on simulated data, it differs a lot from the true Hessian on most applied problems. As a consequence, quasi-Newton algorithms which model the curvature of the objective function with such an approximation can have poor convergence rates. We advocate the L-BFGS method to refine 'on the fly' the approximation of the Hessian. This is supported by experiments on 3 types of real signals which clearly demonstrate that this approach leads to faster convergence.

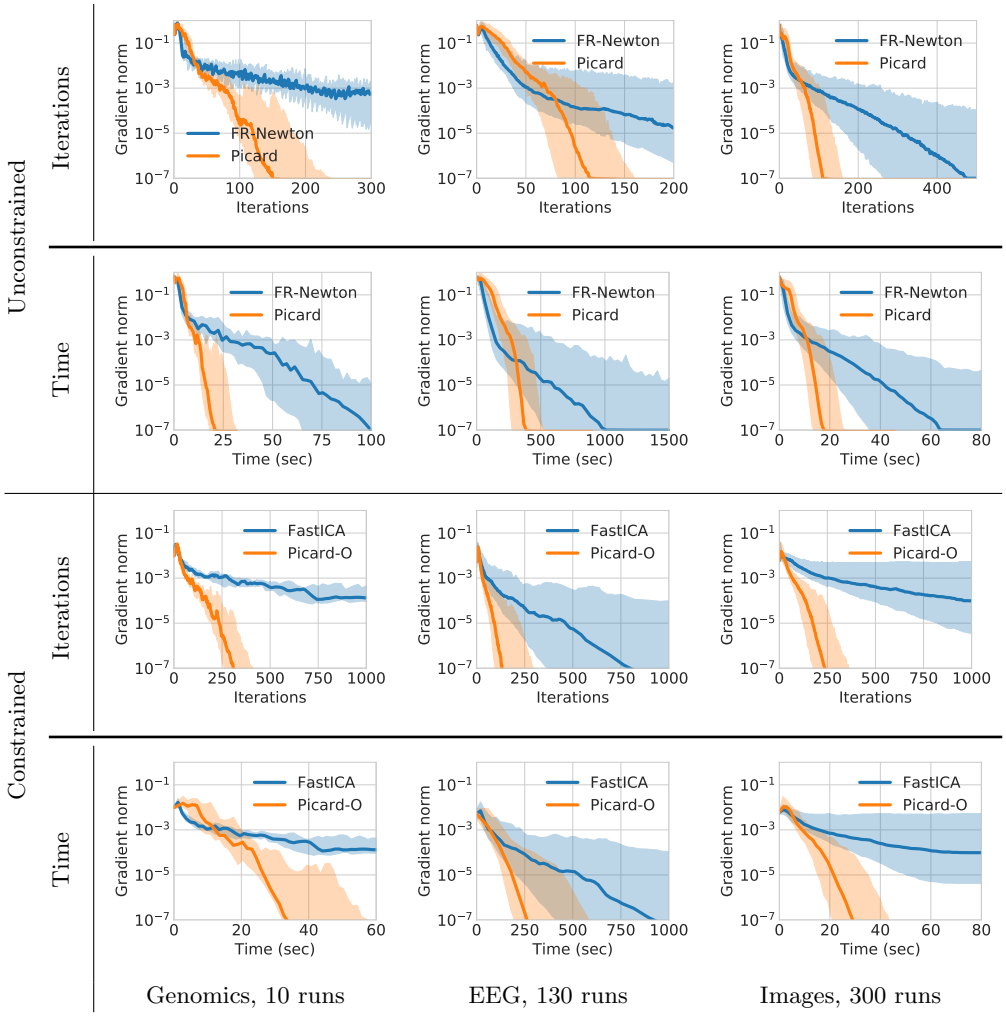

Fig. 2. Convergence speed of several ICA algorithms on 3 real data sets. Each column corresponds to a type of data. The first two rows correspond to the unconstrained algorithms, the last two to the constrained algorithms. The first row of each pair displays the evolution of gradient across iterations, the second one displays the evolution of gradient against time. Bold lines correspond to the medians of the gradient norms, and the shading displays the $10-90\%$ percentile.

References

1. P. Comon, "Independent component analysis, a new concept?" *Signal Processing*, vol. 36, no. 3, pp. 287 – 314, 1994.
2. J. Himberg, A. Hyvärinen, and F. Esposito, "Validating the independent components of neuroimaging time series via clustering and visualization," *NeuroImage*, vol. 22, no. 3, pp. 1214 – 1222, 2004.
3. A. J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural computation*, vol. 7, no. 6, pp. 1129–1159, 1995.
4. J.-F. Cardoso, "Infomax and maximum likelihood for blind source separation," *IEEE Signal processing letters*, vol. 4, no. 4, pp. 112–114, 1997.
5. M. Zibulevsky, "Blind source separation with relative newton method," in *Proc. ICA*, vol. 2003, 2003, pp. 897–902.
6. H. Choi and S. Choi, "A relative trust-region algorithm for independent component analysis," *Neurocomputing*, vol. 70, no. 7, pp. 1502–1510, 2007.
7. J. A. Palmer, K. Kreutz-Delgado, and S. Makeig, "AMICA: An adaptive mixture of independent component analyzers with shared components," Tech. Rep., 2012.
8. A. Hyvärinen, "Fast and robust fixed-point algorithms for independent component analysis," *IEEE Transactions on Neural Networks*, vol. 10, no. 3, pp. 626–634, 1999.
9. ——, "The fixed-point algorithm and maximum likelihood estimation for independent component analysis," *Neural Processing Letters*, vol. 10, no. 1, pp. 1–5, 1999.
10. P. Ablin, J.-F. Cardoso, and A. Gramfort, "Faster ICA under orthogonal constraint," in *Proc. IEEE ICASSP*, 2018.
11. ——, "Faster independent component analysis by preconditioning with hessian approximations," *Arxiv Preprint*, 2017.
12. D. T. Pham and P. Garat, "Blind separation of mixture of independent sources through a quasi-maximum likelihood approach," *IEEE Transactions on Signal Processing*, vol. 45, no. 7, pp. 1712–1725, 1997.
13. J.-F. Cardoso and B. H. Laheld, "Equivariant adaptive source separation," *IEEE Transactions on Signal Processing*, vol. 44, no. 12, pp. 3017–3030, 1996.
14. J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, 1999.
15. H. Karimi, J. Nutini, and M. Schmidt, "Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2016, pp. 795–811.
16. J. Nocedal, "Updating quasi-newton matrices with limited storage," *Mathematics of computation*, vol. 35, no. 151, pp. 773–782, 1980.
17. A. Delorme, J. Palmer, J. Onton, R. Oostenveld, and S. Makeig, "Independent EEG sources are dipolar," *PloS one*, vol. 7, no. 2, p. e30135, 2012.