INFOMAX-ICA USING HESSIAN-FREE OPTIMIZATION

Philippe Tillet, H. T. Kung, David Cox

John A. Paulson School of Engineering and Applied Sciences
Department of Computer Science
Harvard University

ABSTRACT

We present HF-ICA, a second-order "Hessian-free" algorithm for Infomax-ICA. Our approach achieves asymptotically quadratic convergence while retaining the memory footprint of first-order methods. Without any hyperparameter tuning, we show better convergence properties than both other approximate Newton-type methods and finely-tuned stochastic Natural Gradient Descent on EEG and fMRI data. A portable, multi-threaded and vectorized C++ implementation is made publicly available along with MATLAB and Python interfaces.

Index Terms— Infomax-ICA, Hessian-Free Optimization

1. INTRODUCTION

Blind source separation by Independent Component Analysis (ICA) is a standard signal processing algorithm whose applications range from speech enhancement to biomedical signal processing (EEG, MEG, MRI, 2-Photon Microsopy, etc.). In its most basic formulation, one observes a set of random variables $\mathbf{x} = [x_1, \cdots, x_M]$ that are assumed to be linear combinations of as many *latent* and *statistically independent* source signals $\mathbf{s} = [s_1, \cdots, s_M]$:

$$\mathbf{x} = A\mathbf{s}$$

where $A \in \mathbb{R}^{M\times M}$ is an unknown, invertible mixing matrix. The purpose of ICA is to estimate $\bar{W} = A^{-1}$ given $\mathbf{x}$ only.

Different measures of statistical independence lead to different variants of ICA. In this paper, we restrict ourselves to Infomax ICA [1], in which the mutual information (i.e., negative likelihood [2]) of the source signals is minimized:

$$\bar{W} = \arg\min_W L_S(W)$$

$$L_S(W) = -\log|\det W| - \sum_{m=0}^{M}\sum_{n=0}^{N}\log p_m(X_{m,n}) \quad (1)$$

Where N occurences of $\mathbf{x}$ are stacked row-wise to form a data-matrix $X = AS \in \mathbb{R}^{M\times N}$.

Negative log-likelihood minimization for ICA is usually carried out via *Stochastic* Natural Gradient Descent (SNGD). While this method possesses appealing theoretical properties [3], getting good practical performance often requires heavy hyperparameters tuning (e.g., learning rate, annealing factor, mini-batch size). Existing second-order batch methods (Newton's method, Relative ICA [4]) are either too memory intensive or converge too slowly to offset their lack of hyperparameters.

1.1. Contributions

Let $\nabla(W)$ and $H(W)$ be the gradient and the Hessian of the negative log-likelihood L_S at W . Our paper makes the following contributions:

(a) An exact formula for $\mathbf{y} = H(W)\mathbf{v}$ $\quad$ (for any $\mathbf{v} \in \mathbb{R}^M$)

(b) An empirical estimator for $\Sigma = \mathrm{var}_{\mathbf{s}\in S}\nabla_{\mathbf{s}}(W)$

(c) HF-ICA: A memory-efficient 2nd-order algorithm for Infomax ICA based on (a), plus a mini-batch selection procedure based on (b).

(d) A numerical evaluation of Relative Trust Region (RTR)-ICA, SNGD and HF-ICA on EEG and fMRI data.

1.2. Related Work

Several Newton-type methods for Infomax ICA have been developed by the scientific community over the past decade. The exact Newton's method [5] has a prohibitive $\mathcal{O}(M^4)$ memory footprint. Alternative methods that rely on diagonal [4] or block-diagonal [6] approximate Hessians – while better than batch NGD – are in practice much slower to converge than finely tuned SNGD. The method presented in this paper, HF-ICA, uses implicit and arbitrarily accurate Hessian computations to achieve fast convergence while retaining tractable memory consumption.

1.3. Organization

Section 2 provides background and intuition on Amari's Natural Gradient and Hessian-free optimization. Section 3 describes the HF-ICA algorithm. Section 4 shows a variety of

tricks that can be used to further speed up the convergence of HF-ICA. Section 5 numerically evaluates the presented approach. Section 6 provides concluding remarks.

2. BACKGROUND

Let $\mathbf{w} = \text{vec}(W)$ be the vector obtained by stacking the rows of W together. We also define $W = \text{mat}(\mathbf{w})$.

Iterative optimization methods construct a sequence of iterates $\mathbf{w}_t$ that converges to a local/global minimum of the objective function L:

$$\begin{aligned} \mathbf{w}_{t+1} &= \mathbf{w}_t + \alpha_t \mathbf{d}_t \\ \lim_{t\to\infty} \frac{\partial L}{\partial \mathbf{w}_t} &= 0 \end{aligned} \tag{2}$$

Where α_t and $\mathbf{d}_t$ are respectively a step-size and a descent direction chosen at time t. Different choices for these quantities lead to different optimization methods.

2.1. Natural Gradient Descent

The ordinary gradient of L at $\mathbf{w}_t$ is the direction of steepest ascent *per change in L2-norm*:

$$\nabla(\mathbf{w}_t) \propto \underset{\partial\mathbf{w}_t:\text{L2-DIST}(\mathbf{w}_t,\mathbf{w}_t+\partial\mathbf{w}_t)<\epsilon}{\arg\max} L(\mathbf{w}_t + \partial\mathbf{w}_t)$$

Following the ordinary gradient therefore makes the implicit assumption that candidate solutions close to each other (in the L2 sense) have similar semantics (e.g., induce similar probability distributions on the observed data). There is no reason to believe that this is true for maximum-likelihood estimation.

On the other hand, the natural gradient $\tilde{\nabla}$ of L at $\mathbf{w}_t$ is the direction of steepest ascent *per change in KL-divergence*:

$$\tilde{\nabla}_t \propto \underset{\partial\mathbf{w}_t:KL(P(\mathbf{w}_t),P(\mathbf{w}_t+\partial\mathbf{w}_t))<\epsilon}{\arg\max} L(\mathbf{w}_t + \partial\mathbf{w}_t)$$

Where $P(\mathbf{w})$ is the distribution on the recovered source signals under the unmixing matrix induced by $\mathbf{w}$.

Choosing $\mathbf{d}_t = \tilde{\nabla}_t$ for (1) yields the Natural Gradient Descent (NGD) algorithm, which has an elegant formulation for Infomax-ICA [3]:

$$\begin{aligned} W_{t+1} &= W_t - \alpha_t(I - \phi(S_t)S_t^T)W_t \\ \phi(s_{m,n}) &= -\frac{\partial \log p(s_{m,n})}{\partial s_{m,n}} \end{aligned}$$

Here, $S_t = WX_t$, where X_t denotes a subset (i.e., mini-batch) of X chosen at time t. NGD indeed retains convergence in a stochastic setting.

2.2. Hessian-free Optimization

In Newton's method, $\mathbf{d}_t = \mathbf{n}_t$ minimizes the local quadratic approximation of L at $\mathbf{w}_t$:

$$\begin{aligned} \mathbf{n}_t &= \arg\min_{\mathbf{d}}(L(\mathbf{w}_t) + \nabla(\mathbf{w}_t)^T\mathbf{d} + \frac{1}{2}\mathbf{d}^T H(\mathbf{w}_t)\mathbf{d}) \\ \mathbf{n}_t &= -H(\mathbf{w}_t)^{-1}\nabla(\mathbf{w}_t) \\ \alpha_t &= 1 \end{aligned}$$

Note that $H(\mathbf{w}_t) \in \mathbb{R}^{M^2\times M^2}$ is usually too big to fit in memory, hence the emergence of diagonal and block-diagonal approximations.

Alternatively, $\mathbf{n}_t$ can be obtained by solving the linear system:

$$H(\mathbf{w}_t)\mathbf{n}_t = -\nabla(\mathbf{w}_t) \tag{3}$$

The key insight of Hessian-free optimization [7] is that solving this system iteratively (using e.g., the conjugate gradient method) does not require the explicit knowledge of $H(\mathbf{w}_t)$ but only that of $\mathbf{y}_t^v = H(\mathbf{w}_t)\mathbf{v}$ for $\mathbf{v} \in \mathbb{R}^M$.

Note that the $\mathcal{O}(\epsilon)$ approximation:

$$\mathbf{y}_t^v \approx \frac{\nabla(\mathbf{w}_t + \epsilon\mathbf{v}) - \nabla(\mathbf{w}_t)}{\epsilon} \qquad 0 < \epsilon \ll 1 \tag{4}$$

is rarely accurate enough for practical use. In the following section, we derive the exact form of $\mathbf{y}_t^v$.

3. HESSIAN-FREE ICA

Note that (4) evolves into a derivative when $\epsilon \to 0$:

$$\mathbf{y}_t^v \overset{def}{=} \frac{\partial}{\partial\epsilon}\nabla(\mathbf{w}_t + \epsilon\mathbf{v})|_{\epsilon=0} = \mathcal{R}_{\mathbf{v}}\{\nabla(\mathbf{w}_t)\}$$

Different values of $\mathbf{v}$ lead to different operators $\mathcal{R}_{\mathbf{v}}\{.\}$ (we fix $\mathbf{v}$ and drop the subscript from now on). As a differential operator, $\mathcal{R}\{.\}$ obeys the following rules [8]:

$$\begin{aligned} \mathcal{R}\{\mathbf{w}\} &= \mathbf{v} \\ \mathcal{R}\{f(\mathbf{w}) + g(\mathbf{w})\} &= \mathcal{R}\{f(\mathbf{w})\} + \mathcal{R}\{g(\mathbf{w})\} \\ \mathcal{R}\{f(g(\mathbf{w}))\} &= f'(g(\mathbf{w})) \circ \mathcal{R}\{g(\mathbf{w})\} \end{aligned}$$

Let $\{W_t, V, Y_t^V\} = \text{mat}(\{\mathbf{w}_t, \mathbf{v}, \mathbf{y}_t^{\mathbf{v}}\})$. Hessian-vector products for Infomax-ICA can then be computed exactly:

$$\begin{aligned} Y_t^V &= \mathcal{R}\{\nabla L(W)\} \\ Y_t^V &= \mathcal{R}\{W_t^{-1})^T - \phi(S_t)X_t^T\} \\ &= \mathcal{R}\{(W_t^{-1})^T\} - \mathcal{R}\{\phi(S_t)X_t^T\} \\ &= -(W_t^{-1}VW_t^{-1})^T - \psi(S_t)X_t^T \end{aligned} \tag{5}$$

With

$$\begin{aligned}\psi(S_t) &= \mathcal{R}\{\phi(S_t)\} \\ &= \phi'(S_t) \circ \mathcal{R}\{WX_t\} \\ &= \phi'(S_t) \circ VX_t\end{aligned}$$

Where $\circ$ and ϕ' are respectively element-wise multiplication and element-wise differentiation operators. Common choices for ϕ are:

$$\phi(s_{m,n}) = \tanh(s_{m,n})$$
$$\phi(s_{m,n}) = s_{m,n} + \text{sign}(\text{kurtosis}(\mathbf{s}_m)) \tanh(s_{m,n})$$

Giving rise to respectively the original [1] and extended [9] Infomax ICA algorithm.

The linear system (3) can be solved using any existing iterative solver. To keep HF-ICA simple, we used the Conjugate Gradient (CG) method, aborting the procedure whenever a direction of negative curvature was found (CG requires $H(\mathbf{w}_t)$ to be positive definite). There exists a variety of solvers that can deal with indefinite Hessians, such as BiCG-STAB or GMRES.

4. SPEEDING-UP HF-ICA

4.1. Stopping Criterion for CG

Solving the above linear system iteratively yields a solution $\tilde{\mathbf{n}}_t$ which can be made arbitrarily close to the true Newton direction $\mathbf{n}_t$. In practice, however, it could be desirable to avoid Newton's exact direction – $H(\mathbf{w}_t)$ can be indefinite – or unnecessary CG iterations.

We tried stopping criterions based on Hessian-vector products variance [10], quadratic approximation minimization [7], and residual norms. The latter seemed to work best:

$$\tilde{\mathbf{n}}_t^T H \tilde{\mathbf{n}}_t < 0 \qquad \text{(negative curvature)}$$
$$\text{or} \quad \|H\tilde{\mathbf{n}}_t + \nabla(\mathbf{w}_t)\|_2 < \epsilon = 10^{-3} \qquad \text{(convergence reached)}$$

4.2. Damping

Since H can be indefinite, it is common to solve instead:

$$(H(\mathbf{w}_t) + \lambda_t I)\mathbf{n}_t = -\nabla(\mathbf{w}_t)$$

Where λ_t is a damping parameter, updated across iterations as follow:

$$\lambda_{t+1} = (2/3)\lambda_t \qquad \text{if } \rho_t > 0.75$$
$$\lambda_{t+1} = (3/2)\lambda_t \qquad \text{if } \rho_t < 0.25$$

With

$$\rho_t = \frac{L(\mathbf{w}_{t+1}) - L(\mathbf{w}_t)}{q_t^{(last)} - q_t^{(0)}}$$

Where $q_t^{(0)}$ and $q_t^{(last)}$ denote the value of the local quadratic model of L at respectively the first and last iteration of CG.

4.3. Line-search

The true Newton direction $\mathbf{n}_t$ is naturally well scaled. This may not be the case of $\tilde{\mathbf{n}}_t$ when H_t is indefinite or ϵ is large. Many methods exist for finding quasi-optimal step-sizes:

$$\alpha_t = \arg\min_\alpha g(\alpha) = \arg\min_\alpha L(\mathbf{w}_t + \alpha\tilde{n}_t)$$

We chose one based on polynomial interpolations of g (see [11] for more details). This procedure be not often needed in practice (i.e., $\tilde{\mathbf{n}}_t$ is generally close to $\mathbf{n}_t$).

4.4. Adaptive mini-batch size

We've been until now rather vague on how to choose the mini-batch size $|X_t|$. Choosing it naively can break the convergence of Hessian-free algorithms. Instead, Byrd et al. [10] suggest to select $|X_t|$ based on $\Sigma_t = \text{var}_{\mathbf{s}\in S}\nabla_{\mathbf{s}}(\mathbf{w}_t)$. Specifically, this mini-batch size is augmented iff:

$$\frac{\|\Sigma_t\|_1}{|X_t|} \leq (\theta\|\nabla(\mathbf{w}_t)\|_2)^2$$

and becomes

$$|X_{t+1}| = \frac{\|\Sigma_t\|_1}{(\theta\|\nabla(\mathbf{w}_t)\|_2)^2}$$

With:

$$\begin{aligned}\Sigma_t &= \frac{1}{|X_t|-1}\sum_{\mathbf{x}\in X_t}\Big(\phi(\mathbf{s})\mathbf{x}^T - \frac{\phi(S_t)X_t^T}{|X_t|}\Big)^{\circ\circ} \\ &= \frac{1}{|X_t|-1}\Big(\phi(S_t)^{\circ\circ}X_t^{T\circ\circ} - \frac{(\phi(S_t)X_t^T)^{\circ\circ}}{|X_t|}\Big)\end{aligned}$$

Where $^{\circ\circ}$ denotes element-wise squaring.

5. NUMERICAL EXPERIMENTS

It has been suggested that NGD could be more theoretically sound than Newton-type methods for learning tasks, while the latter could achieve faster convergence in optimization problems [12]. In this section, we compare the quality of the local minima found by various optimization methods (SNGD, RTR-ICA, HF-ICA) on EEG and fMRI data. FAST-ICA was not included because it maximizes likelihood only approximately [13]. Whether or not achieving lower mutual information (i.e., higher maximum-likelihood) for the recovered sources translates to better separation in practice is beyond the scope of this paper.

Two different implementations of SNGD are evaluated:

(a) Heuristics: the learning rate and annealing factor follow heuristics provided in the EEGLAB [14] software:

$$\text{lrate} = \frac{6.5 \times 10^{-4}}{\log_e(N_{IC})}$$
$$\text{anneal} = 0.9$$

(b) Fine-Tuned: the learning rate and annealing factor are found using a grid-search:

$$\text{lrate} \in \{10^{-4}, 2\times10^{-4}, \cdots, 9\times10^{-4}\}$$
$$\text{anneal} \in \{0.85, \cdots, 0.91, \cdots, 0.99\}$$

The configuration achieving the best local minimum is retained.

We report here the results observed for Infomax-ICA. The same behaviors were observed for the extended algorithm (also implemented in our software package).

5.1. EEG

60-channels motor imagery EEG data were obtained from the BCI Competition IV datasets [1]. Pre-whitening and dimensionality reduction via PCA was applied to retain 99.9% of the data variance, and ICA was ran on the resulting $29 \times 190,594$ data-points.

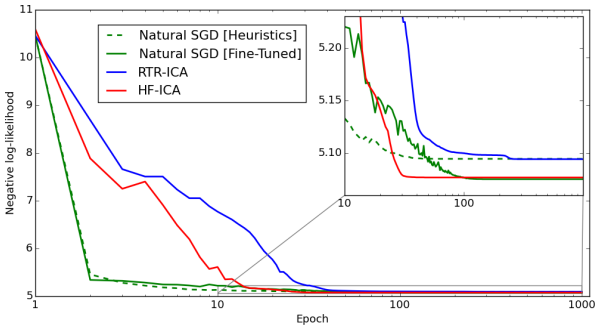

Fig. 1. Convergence speed of HF-ICA on EEG data

As shown in Figure 1, RTR-ICA and non-tuned NSGD converge to the same sub-optimal local minimum. HF-ICA reaches the same optimum as Fine-tuned NSGD faster (50 vs 117 epochs).

5.2. fMRI

We used publicly available fMRI data obtained from one human subject during a language task [2] (316 frames of $104 \times 90 \times 72$ voxels). Pre-whitening and dimensionality reduction was applied to retain 99.9% of the data variance, and ICA was ran on the resulting $167 \times 673,920$ data-points.

As shown in Figure 2, SNGD reaches a poor local optimum when not properly tuned. Fine-tuned NSGD and HF-ICA again reaches the same local optimum as NSGD faster (600 vs 800 iterations). RTR-ICA didn't reach convergence after 10,000 iterations.

[1] http://www.bbci.de/competition/iv/desc_1.html
[2] http://humanconnectome.org

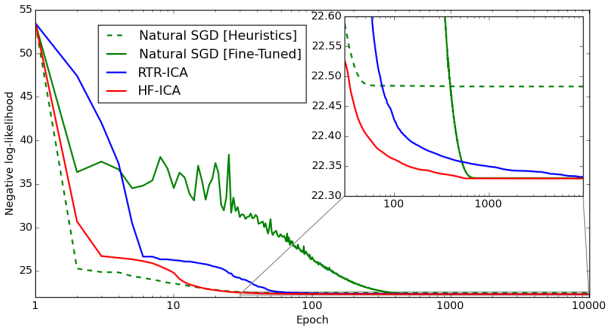

Fig. 2. Convergence speed of HF-ICA on fMRI data

5.3. Execution time

Table 5.3 summarized the runtime of the aforementioned benchmarks. The same stopping criterion was used in all cases: $\|\mathbf{w}_{t+1} - \mathbf{w}_t\|_2 < 10^{-4}$.

	RTR-ICA	HF-ICA	SNGD (Tuned)	Grid-search
EEG	43s	20s	66s	1h30m
fMRI	N/A	6h	2h	>60h

Table 1. Runtime of HF-ICA

Note that each iteration of HF-ICA can involve many Hessian-vector products computations and be substantially more expensive than each epoch of NSGD. The additional parallelism induced by larger batch sizes mitigates this effect. CG preconditioning [15] could further improve HF-ICA.

6. CONCLUSIONS

In this paper, we presented a second-order method for Infomax ICA based on hessian-Free optimization. Performance similar to SNGD and superior to Relative ICA are observed – without requiring any hyperparameter tuning. The proposed approach is safe for practical use and an optimized package is available for C++, Python and Matlab [3].

[3] https://github.com/ptillet/hf-ica

7. REFERENCES

[1] A. J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Comput.*, 1995.

[2] J.-F. Cardoso, "Infomax and maximum likelihood for blind source separation," *IEEE Letters on Signal Processing*, 1997.

[3] S.-I. Amari, "Natural gradient works efficiently in learning," *Neural Comput.*, 1998.

[4] H. Choi and S. Choi, "A relative trust-region algorithm for independent component analysis," *Neurocomput.*, 2007.

[5] J. A. Palmer, S. Makeig, K. Kreutz-Delgado, and B. D. Rao, "Newton method for the ica mixture model," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, 2008.

[6] A. M. Bronstein, M. M. Bronstein, and M. Zibulevsky, "Blind source separation using the block-coordinate relative newton method," in *Independent Component Analysis and Blind Signal Separation, Fifth International Conference, ICA 2004*, 2004.

[7] J. Martens, "Deep learning via hessian-free optimization," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010.

[8] B. A. Pearlmutter, "Fast exact multiplication by the hessian," *Neural Comput.*, 1994.

[9] T.-W. Lee, M. A. Girolami, and T. J. Sejnowski, "Independent component analysis using an extended infomax algorithm for mixed sub-gaussian and super-gaussian sources," *Neural Computation*, 1999.

[10] R. H. Byrd, G. M. Chin, J. Nocedal, and Y. Wu, "Sample size selection in optimization methods for machine learning," *Math. Program.*, 2012.

[11] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, 2nd edition, 2006.

[12] Nicolas Le Roux and Andrew W. Fitzgibbon, "A fast natural newton method," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010.

[13] A. Hyvärinen, "The fixed-point algorithm and maximum likelihood estimation for independent component analysis," *Neural Processing Letters*, 1999.

[14] A. Delorme and S. Makeig, "Eeglab: an open source toolbox for analysis of single-trial eeg dynamics including independent component analysis," *Journal of Neuroscience Methods*, 2004.

[15] O. Chapelle and D. Erhan, "Improved preconditioner for hessian free optimization," in *In NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.