Accelerating likelihood optimization for ICA on real signals

Pierre Ablin
INRIA

Joint work with: JF. Cardoso & A. Gramfort

LVA-ICA 2018

Motivation

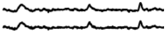
Standard linear ICA solvers, e.g. Infomax/FastICA, are widely used in applied science.

Slow convergence on real data

- Understand why?
- Provide faster algorithms

Maximum likelihood ICA

The linear ICA model

Observations: N signals of length T, $X \in \mathbb{R}^{N \times T}$ 

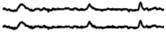
Generative model: There exists a matrix $A \in \mathbb{R}^{N \times N}$ and independent signals $[s_1, \cdots, s_N]^\top = S \in \mathbb{R}^{N \times T}$ such that:

$$\boxed{X = AS}$$

White signals :

We assume $C_X = \frac{1}{T} X X^\top = I_N$ (decorrelated signals).
Enforce it by $X \leftarrow C_X^{-1/2} X$

The linear ICA model

Observations: N signals of length T, $X \in \mathbb{R}^{N \times T}$ 

Generative model: There exists a matrix $A \in \mathbb{R}^{N \times N}$ and independent signals $[s_1, \cdots, s_N]^\top = S \in \mathbb{R}^{N \times T}$ such that:

$$\boxed{X = AS}$$

White signals :

We assume $C_X = \frac{1}{T} X X^\top = I_N$ (decorrelated signals).

Enforce it by $X \leftarrow C_X^{-1/2} X$

Likelihood of the model

Density of the sources: $s_i \sim p_i$.

Likelihood of the model:

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

Cost function: $\mathcal{L}(W) = -\frac{1}{T}\log(p(X|W^{-1}))$

$$\boxed{\mathcal{L}(W) = -\log|\det W| + \sum_{i=1}^{N} \hat{E}[-\log(p_i([WX]_{it}))]}$$

Maximum likelihood ICA

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

- Find $W = \arg\min \mathcal{L}(W)$ (maximum likelihood estimator)
- Solved by Infomax[1] with fixed densities ($\forall i, p_i = p$)

Orthogonal constraint:

- Find $W = \arg\min \mathcal{L}(W)$ subject to $WW^\top = I_N$.
- Solved by Fastica[2] with a binary switch between densities ($\forall i, \log(p_i) = \pm \log(p)$)

[1] Bell, Sejnowski, "An information-maximization approach to blind separation and blind deconvolution", 1995

[2] Hyvarinen, "Fast and robust fixed-point algorithms for independent component analysis", 1999

Maximum likelihood ICA

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

- Find $W = \arg\min \mathcal{L}(W)$ (maximum likelihood estimator)
- Solved by Infomax[1] with fixed densities ($\forall i, p_i = p$)

Orthogonal constraint:

- Find $W = \arg\min \mathcal{L}(W)$ subject to $WW^\top = I_N$.
- Solved by Fastica[2] with a binary switch between densities ($\forall i, \log(p_i) = \pm \log(p)$)

[1] Bell, Sejnowski, "An information-maximization approach to blind separation and blind deconvolution", 1995

[2] Hyvarinen, "Fast and robust fixed-point algorithms for independent component analysis", 1999

An optimization problem

Geometry of the cost function

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

- Optimization on the set of invertible matrices
- Non-Convex problem

Relative (multiplicative) update:

$$\boxed{W \leftarrow \exp(\mathcal{E})W,\ \mathcal{E} \in \mathbb{R}^{N\times N}}$$

- W remains invertible
- Easy to enforce orthogonal constraint: take $\mathcal{E}$ antisymmetric

Derivatives of the cost function

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

Second order expansion:

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

$G \in \mathbb{R}^{N\times N}$, $H \in \mathbb{R}^{N\times N\times N\times N}$

Define $\psi_i(\cdot) = -\log(p_i(\cdot))' = -\frac{p_i'(\cdot)}{p_i(\cdot)}$, $Y = WX$.

- $G_{ij} = \hat{E}[\psi_i(y_i)y_j] - \delta_{ij}$ ($\delta_{ij} = 1$ if $i = j$, 0 else)
- $H_{ijkl} = \delta_{il}\delta_{jk}\hat{E}[\psi_i(y_i)y_i] + \delta_{ik}\,\hat{E}[\psi_i'(y_i)y_jy_l]$

Derivatives of the cost function

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

Second order expansion:

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

$G \in \mathbb{R}^{N\times N}$, $H \in \mathbb{R}^{N\times N\times N\times N}$

Define $\psi_i(\cdot) = -\log(p_i(\cdot))' = -\frac{p_i'(\cdot)}{p_i(\cdot)}$, $Y = WX$.

- $G_{ij} = \hat{E}[\psi_i(y_i)y_j] - \delta_{ij}$ ($\delta_{ij} = 1$ if $i = j$, 0 else)
- $H_{ijkl} = \delta_{il}\delta_{jk}\hat{E}[\psi_i(y_i)y_i] + \delta_{ik}\,\hat{E}[\psi_i'(y_i)y_jy_l]$

Newton's method?

$$G_{ij} = \hat{E}[\psi_i(y_i)y_j] - \delta_{ij}$$
$$H_{ijkl} = \delta_{il}\delta_{jk}\hat{E}[\psi_i(y_i)y_i] + \delta_{ik}\,\hat{E}[\psi_i'(y_i)y_jy_l]$$

$$\boxed{\mathcal{E} = -H^{-1}G}$$

$$W \leftarrow \exp(\mathcal{E})W$$

- Quadratic convergence ☺
- H is costly to compute: $O(N^3T)$ ☹
- H is costly to regularize, and invert ☹
- Not practical

Hessian approximation

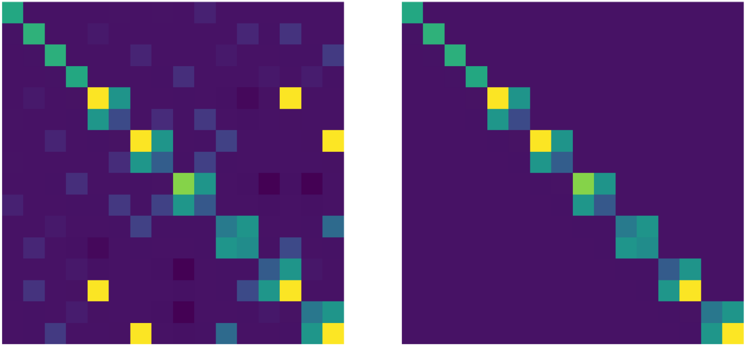
$$H_{ijkl} = \delta_{il}\delta_{jk}\hat{E}[\psi_i(y_i)y_i] + \delta_{ik}\,\hat{E}[\psi_i'(y_i)y_j y_l]$$

If the signals in Y are **independent** and there are **infinitely many samples**, H simplifies:

$$\tilde{H}_{ijkl} = \delta_{il}\delta_{jk}\hat{E}[\psi_i(y_i)y_i] + \delta_{ik}\delta_{jl}\,\hat{E}[\psi_i'(y_i)y_j^2]$$

- Cheaper to compute ($O(N^2T)$, as costly as a gradient) ☺
- *Block diagonal* structure with blocks of size 2
- Easy to regularize (regularize each block) ☺
- Easy to invert (invert each block) ☺

On a 4 sources problem



H $\tilde{H}$

Idea: use $\tilde{H}$ for Newton's method

$$\boxed{\mathcal{E} = -\tilde{H}^{-1}G}$$

$$W \leftarrow \exp(\mathcal{E})W$$

- Fast-relative Newton[3]
- FastICA follows similar iterations with projection [4]:

$$\mathcal{E} \leftarrow \frac{\mathcal{E} - \mathcal{E}^\top}{2}$$

Key remark: $\tilde{H}$ is a good approximation only when the signals are independent...

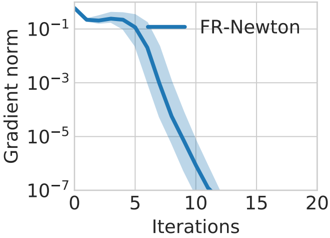
[3] Zibulevski, "Blind source separation with relative newton method", 2003

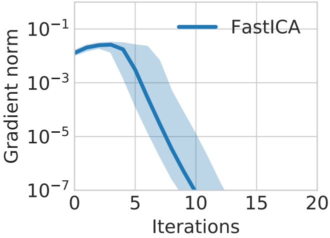
[4] Ablin et al., "Faster ICA under orthogonal constraint", 2018

Practical example

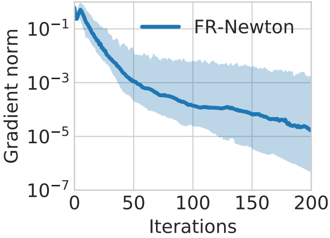
Synthetic data $\neq$ real data

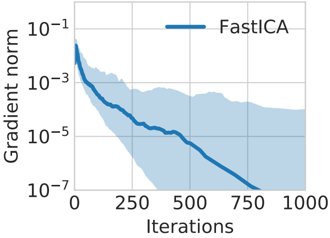
- $N = 8$ independent sources S, $X = AS$





- $N = 8$ EEG signals, X

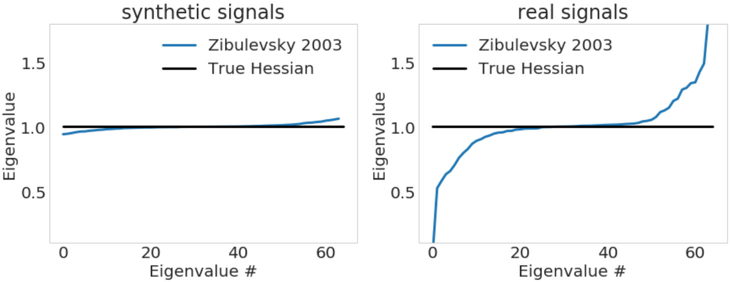




What's going on?

- On the EEG signals, the ICA model $X = AS$ is only true **to some extent**.
- $\tilde{H}$ is never a really good approximation of H

Spectrum of $\tilde{H}^{-\frac{1}{2}} H \tilde{H}^{-\frac{1}{2}}$:



Bad conditioning

The Picard algorithm

Preconditioning

- $\tilde{H}$ is not good enough on real signals
- Use $\tilde{H}$ as a preconditioner

L-BFGS is a widely spread quasi-Newton algorithm

- Uses the previous iterations $W_n, W_{n-1}, \cdots$ and gradient values $G_n, G_{n-1}, \cdots$ to build an approximation of H
- No prior knowledge on the problem
- Starts from an initial guess λI_d in the standard version
- Simply use $\tilde{H}$ as initialization!

Orthogonal constraint: Project $\mathcal{E}$: $\mathcal{E} \leftarrow \frac{\mathcal{E} - \mathcal{E}^\top}{2}$

Preconditioned ICA for Real Data[5]

[5] Ablin et al., "Faster ICA by preconditioning with Hessian approximations", 2017

Preconditioning

- $\tilde{H}$ is not good enough on real signals
- Use $\tilde{H}$ as a preconditioner

L-BFGS is a widely spread quasi-Newton algorithm

- Uses the previous iterations $W_n, W_{n-1}, \cdots$ and gradient values $G_n, G_{n-1}, \cdots$ to build an approximation of H
- No prior knowledge on the problem
- Starts from an initial guess λI_d in the standard version
- Simply use $\tilde{H}$ as initialization!

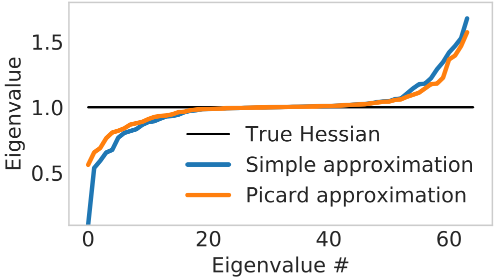
Orthogonal constraint: Project $\mathcal{E}$: $\mathcal{E} \leftarrow \frac{\mathcal{E}-\mathcal{E}^\top}{2}$

Preconditioned **ICA** for **R**eal **D**ata[5]

[5] Ablin et al., "Faster ICA by preconditioning with Hessian approximations", 2017

Better conditioning

Picard's Hessian approximation is built upon $\tilde{H}$, and refined using the past.



Results on real data

Genomics dataset

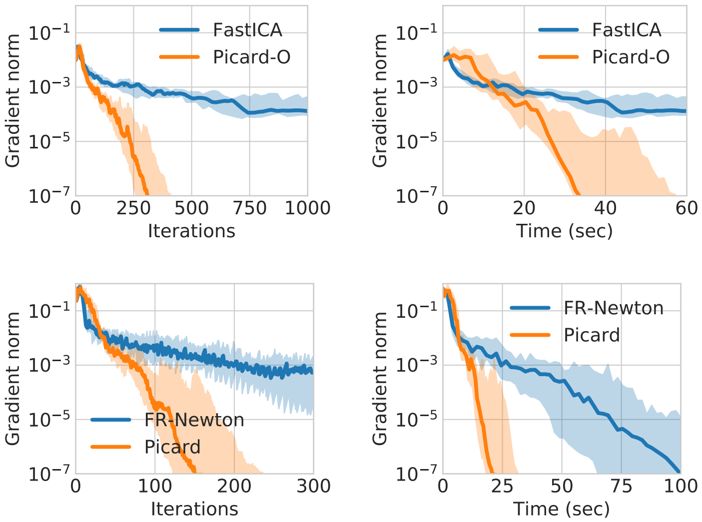
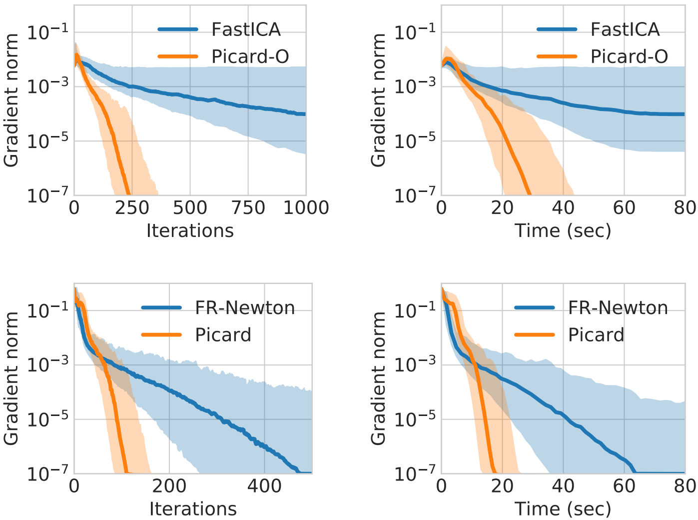
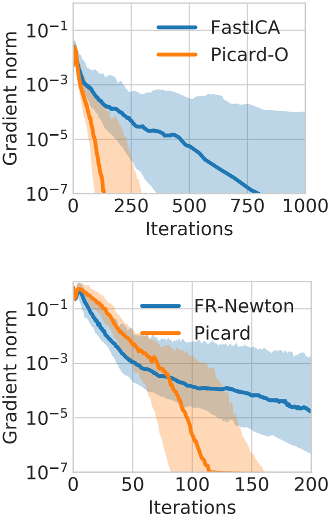


Image patch dataset



EEG dataset



Gradient norm
FastICA
Picard-O
Iterations
Time (sec)
FR-Newton
Picard
10^{-1}
10^{-3}
10^{-5}
10^{-7}
0
250
500
750
1000
50
100
150
200
1500

Conclusion

- Speed of standard algorithms (FastICA, Fast-Relative Newton) critically relies on the independence assumption
- In a realistic setting, this assumption **never really holds**
- The Picard algorithm overcomes this issue, finds the same solutions much faster

Python/Matlab/Octave code available online!

https://github.com/pierreablin/picard

P. Ablin, J. F. Cardoso and A. Gramfort, "Faster ICA by Preconditioning With Hessian Approximations," in *IEEE TSP*, 2018

Thanks for your attention!

Conclusion

- Speed of standard algorithms (FastICA, Fast-Relative Newton) critically relies on the independence assumption
- In a realistic setting, this assumption **never really holds**
- The Picard algorithm overcomes this issue, finds the same solutions much faster

Python/Matlab/Octave code available online!

https://github.com/pierreablin/picard

P. Ablin, J. F. Cardoso and A. Gramfort, "Faster ICA by Preconditioning With Hessian Approximations," in *IEEE TSP*, 2018

Thanks for your attention!