Approximate Message-Passing for Convex Optimization with Non-Separable Penalties

Andre Manoel[†,‡], Florent Krzakala[♯], Bertrand Thirion[†], Gaël Varoquaux[†] and Lenka Zdeborová[‡]

† Parietal Team, Inria, Neurospin, CEA, ‡ Institut de Physique Théorique, CEA and ♯ Laboratoire de Physique Statistique, ENS

Convex optimization with non-separable penalties

▷ **Problem statement**— we consider the minimization of an objective consisting of a quadratic loss and a non-separable penalty

$$\arg\min_{x} \ \frac{1}{2}\|\boldsymbol{y} - A\boldsymbol{x}\|_2^2 + \lambda \sum_{k=1}^{R} f\big((K\boldsymbol{x})_k\big)$$

for given $\boldsymbol{y} \in \mathbb{R}^N$, $A \in \mathbb{R}^{N\times P}$ and $K \in \mathbb{R}^{R\times P}$. Prominent examples are the total variation (TV) penalty and the cosparse analysis model

▷ **Better algorithms?**— proximal algorithms such as FISTA and ADMM are the state-of-the-art for performing this minimization. However, both have issues

- in FISTA, convergence is slow due to the inner loop requiring more and more iterations
- in ADMM, the behavior is highly dependent on the stepsize, which is hard to set

We thus look for alternative approaches that are hopefully faster and/or require less parameters

▷ **Our approach**— promising new class of algorithms: *approximate message-passing* (AMP)

Idea: adapt the vector approximate-message passing (VAMP) algorithm [Rangan et al. 2017] and the expectation-consistent (EC) approximation [Opper and Winther 2005] for non-separable penalties

We rederive the iteration from scratch and benchmark it on standard datasets: promising results!

Approximate message-passing

▷ **Probabilistic framework**— given the following probability distribution

$$P(\boldsymbol{x}|A,\boldsymbol{y}) = \frac{1}{\mathcal{Z}} e^{-\frac{1}{2}\|\boldsymbol{y}-A\boldsymbol{x}\|_2^2} \prod_{j=1}^{P} e^{-\lambda f(x_j)}$$

the AMP algorithm [Donoho et al. 2009] is able to compute the MAP estimator by means of the following iteration

$$\boldsymbol{x}^{t+1} = \eta_{\lambda\sigma^t}(\boldsymbol{x}^t + A^T\boldsymbol{z}^t)$$
$$\boldsymbol{z}^t = \boldsymbol{y} - A\boldsymbol{x}^t + \frac{1}{\alpha}\boldsymbol{z}^{t-1}\big\langle \nabla\eta_{\lambda\sigma^t}(\boldsymbol{x}^t + A^T\boldsymbol{z}^t)\big\rangle$$

where $\eta_\lambda(v) = \mathrm{prox}_{\lambda f}(v)$. For a ℓ_1 penalty, $f(x) = |x|$: soft-thresholding

A lot like ISTA, but w/ an additional term and an adaptive stepsize based on the variance of $\boldsymbol{x}$. Usually faster, however: convergence issues!

▷ **The vector AMP (VAMP) algorithm** [Rangan et al. 2017]— more robust than AMP; MAP estimator comes from

$$\boldsymbol{x}^t = (A^TA + \rho^t I_N)^{-1}(A^T\boldsymbol{y} + \boldsymbol{u}^t), \qquad \boldsymbol{z}^t = \eta_{\lambda\sigma_x^t/(1-\sigma_x^t\rho^t)}\left(\frac{\boldsymbol{x}^t - \sigma_x^t\boldsymbol{u}^t}{1-\sigma_x^t\rho^t}\right),$$
$$\boldsymbol{u}^{t+1} = \boldsymbol{u}^t + (\boldsymbol{z}^t/\sigma_z^t - \boldsymbol{x}^t/\sigma_x^t), \qquad \rho^{t+1} = \rho^t + (1/\sigma_z^t - 1/\sigma_x^t).$$

A lot like ADMM (more precisely, the Peaceman-Rachford splitting) with, once again, an adaptive stepsize based on variance of $\boldsymbol{x}$. Upon convergence, $\boldsymbol{x} = \boldsymbol{z}$

▷ **Some facts about VAMP**—

▷ as in ADMM, A^TA can be replaced by its eigendecomposition, so that matrix inverse is not necessary (however: extra preprocessing costs)

▷ is not able in principle to deal with losses other than quadratic (but can be adapted [He et al. 2017]), nor non-separable penalties

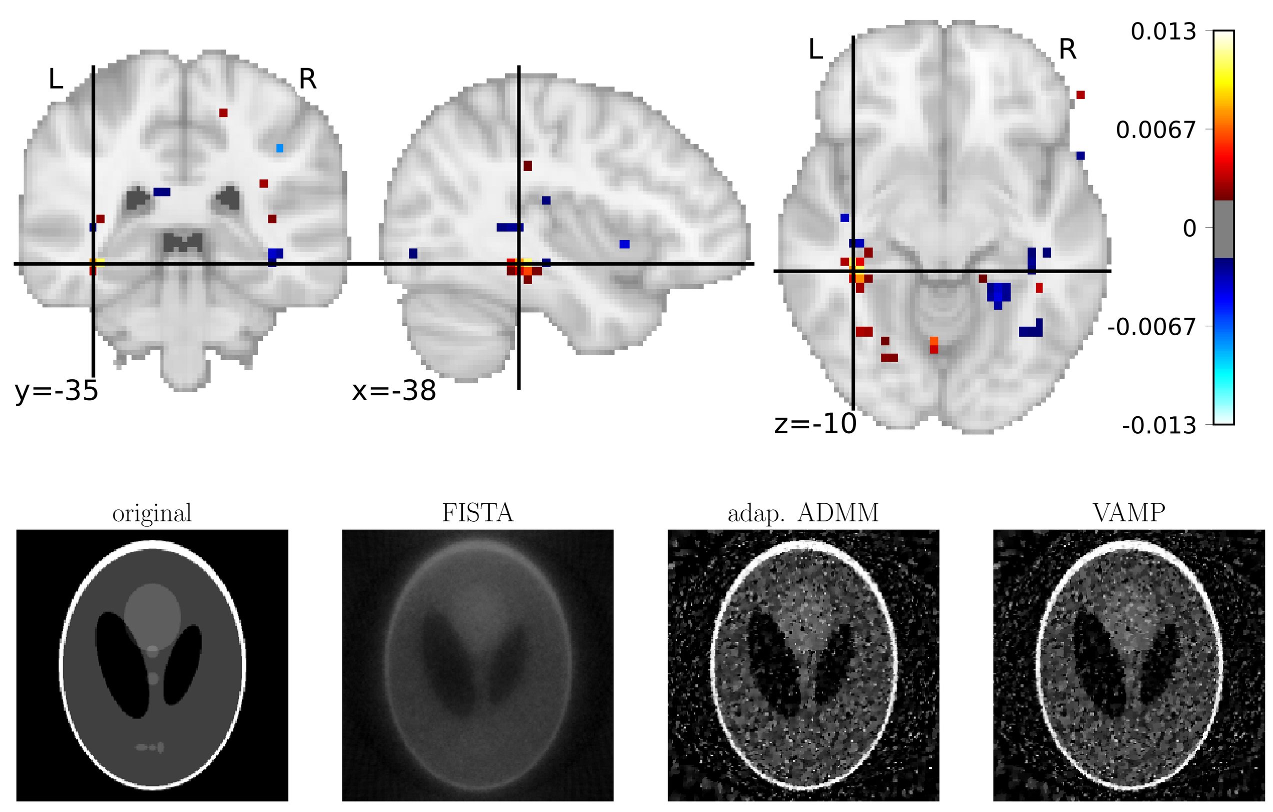


Figure: *Left:* **Sample of results obtained using proposed iteration** with TV penalties. Top: classification on Haxby, final result. Bottom: tomography on the Shepp-Logan phantom (bottom), 10s after preprocessing

Benchmarking the proposed iteration

▷ **Benchmarks**— we use a TV penalty ($K = \nabla$) and approach two problems:

▷ one vs. all classification on task fMRI on the Haxby dataset ($N = 1452$, $P = 136840$), 3 labels: "face", "house" and "chair"

▷ tomography on noisy projections of the Shepp-Logan phantom ($P = 40000$), 1% SNR noise

▷ **Tricks to speed up iteration**— in the $N \ll P$ setting: Woodbury formula, $K^TK = \Delta$ diagonal in Fourier basis (use FFT instead)

▷ **Some inconvenients**— as with PRS, convergence is not assured: relaxation parameter in the updates of $\boldsymbol{u}$ and ρ

▷ **Perspectives**— losses other than quadratic, imposing monotonicity, confidence interval from variance estimates, more experiments

References

[1] Rangan, S., Schniter, P. & Fletcher, A. K. (2017). Vector approximate message passing. In IEEE International Symposium on Information Theory (pp. 1588-1592).

[2] Opper, M. & Winther, O. (2005). Expectation consistent approximate inference. Journal of Machine Learning Research, 6 (Dec), 2177-2204.

[3] Donoho, D. L., Maleki, A. & Montanari, A. (2009). Message-passing algorithms for compressed sensing. *PNAS*.

[4] Manoel, A., Krzakala, F., Thirion, B., Varoquaux, G. & Zdeborová, L. (2018). Approximate message-passing for convex optimization with non-separable penalties. In preparation.

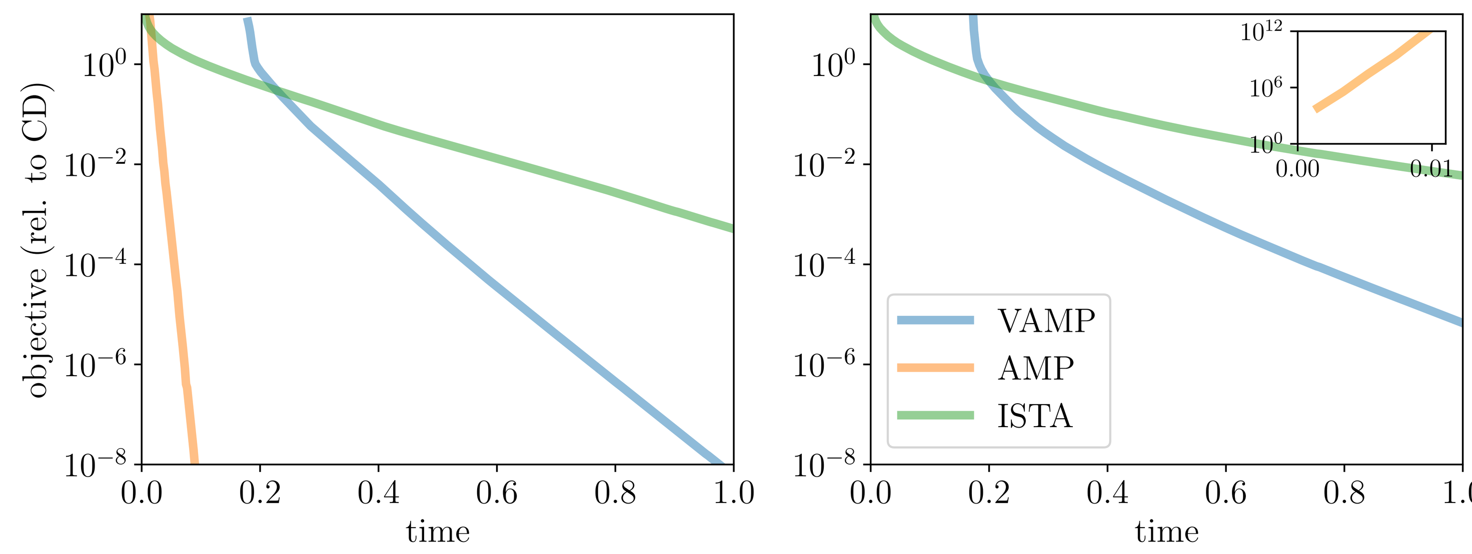


Figure: **Comparison between AMP and VAMP using ℓ_1 regularization** on synthetic data: for i.i.d. Gaussian matrices (left) AMP is faster, but already for products of Gaussian i.i.d. matrices it diverges (right)

Adapting VAMP to non-separable penalties

▷ **The expectation-consistent (EC) approximation** [Opper and Winther 2005]— given a probability distribution

$$P(\boldsymbol{x}) = \frac{1}{\mathcal{Z}} P_\ell(\boldsymbol{x})\, P_r(\boldsymbol{x})$$

the (negative) log-partition function $-\log\mathcal{Z}$ is approximated by

$$\mathcal{F}[Q_\ell, Q_r] = -\log\int d\boldsymbol{x}\, P_\ell(\boldsymbol{x})\, Q_r(\boldsymbol{x}) - \log\int d\boldsymbol{x}\, Q_\ell(\boldsymbol{x})\, P_r(\boldsymbol{x}) + \log\int d\boldsymbol{x}\, Q_\ell(\boldsymbol{x})\, Q_r(\boldsymbol{x})$$

for a tractable choice of $Q_{\ell,r}$, typically

$$Q_{\ell,r}(\boldsymbol{x}) = \exp\left(-\frac{1}{2}\rho_{\ell,r}\boldsymbol{x}^T\boldsymbol{x} + \boldsymbol{u}_{\ell,r}^T\boldsymbol{x}\right)$$

One must then optimize over $\boldsymbol{u}_{\ell,r}$ and $\rho_{\ell,r}$. VAMP performs this optimization via a fixed-point iteration

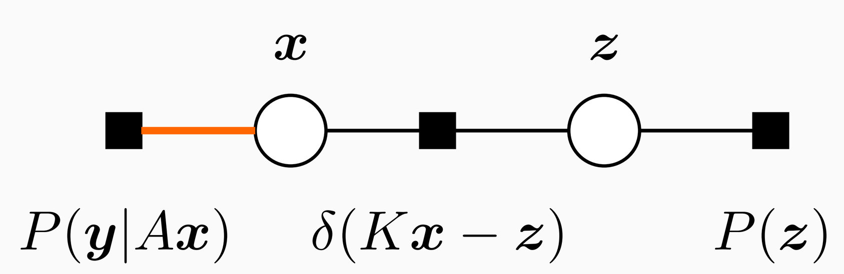
If one considers instead $[P_\ell(\boldsymbol{x})\, P_r(\boldsymbol{x})]^\beta$ and takes the limit $\beta \to \infty$, the MAP estimate is recovered from $\mathbb{E}\boldsymbol{x} = -\nabla_{\boldsymbol{u}_{\ell,r}}\mathcal{F}$

▷ **Adapting to TV**— we introduce a new variable $\boldsymbol{z} = K\boldsymbol{x}$ and use the EC approximation not on $P(\boldsymbol{x}|A,\boldsymbol{y})$ but on

$$P(\boldsymbol{z}|A,\boldsymbol{y}) \propto \int d\boldsymbol{x}\, e^{-\frac{1}{2}\|\boldsymbol{y}-A\boldsymbol{x}\|_2^2}\, \delta(\boldsymbol{z} - K\boldsymbol{x}) \prod_{k=1}^{R} e^{-\lambda f(z_k)}.$$

The following VAMP-like iteration can be derived in this case

$$\boldsymbol{x}^t = (A^TA + \rho^t K^TK)^{-1}(A^T\boldsymbol{y} + K^T\boldsymbol{u}^t) \qquad \boldsymbol{z}^t = \eta_{\lambda\sigma_x^t/(1-\sigma_x^t\rho^t)}\left(\frac{K\boldsymbol{x}^t - \sigma_x^t\boldsymbol{u}^t}{1-\sigma_x^t\rho^t}\right)$$
$$\boldsymbol{u}^{t+1} = \boldsymbol{u}^t + (\boldsymbol{z}^t/\sigma_z^t - K\boldsymbol{x}^t/\sigma_x^t) \qquad \rho^{t+1} = \rho^t + (1/\sigma_z^t - 1/\sigma_x^t)$$



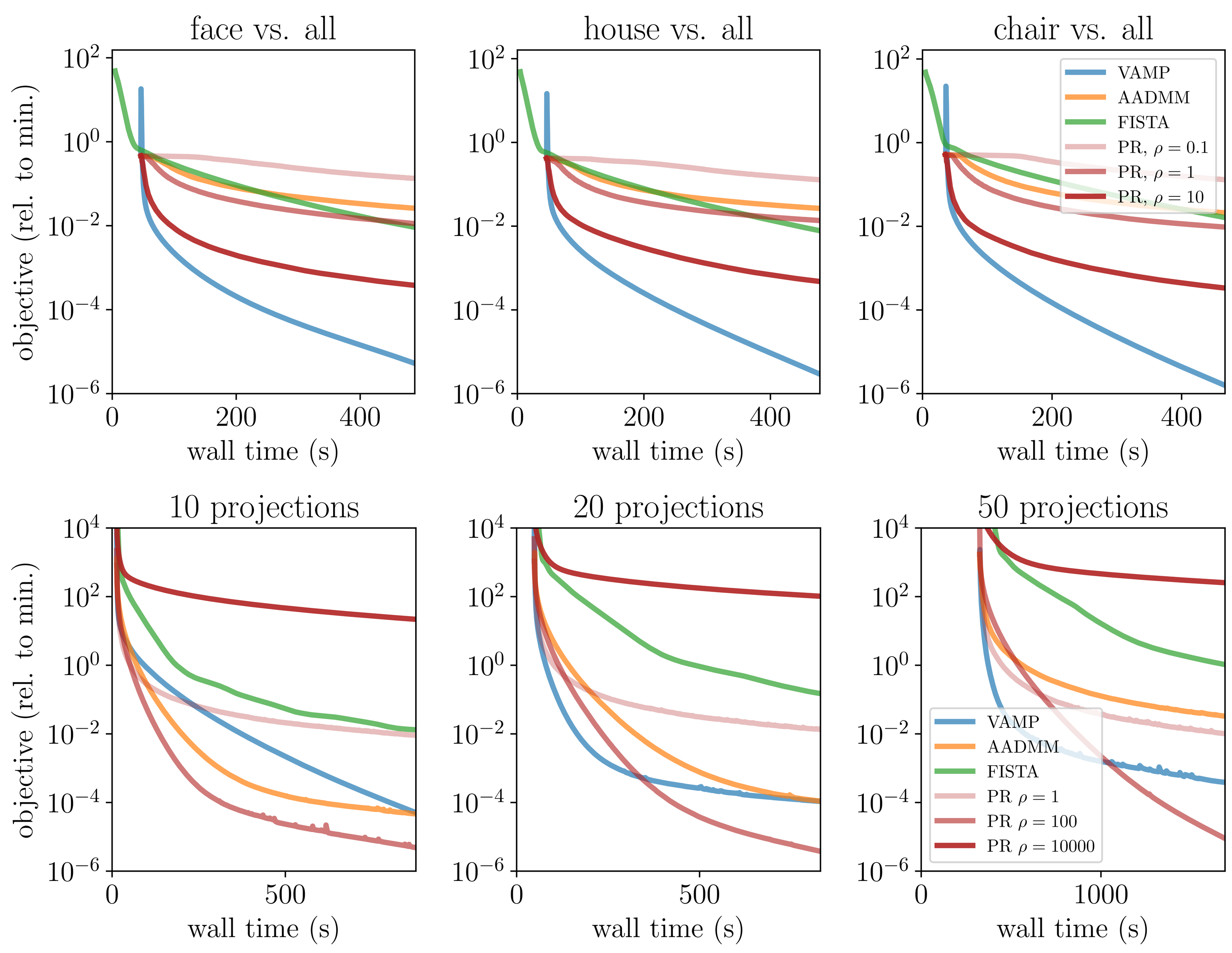


Figure: **Comparison between different approaches using TV penalties**, for classification (top) and tomography (bottom). VAMP is competitive, and often faster than other approaches