# The limited-memory recursive variational Gaussian approximation (L-RVGA)

Marc Lambert, Silvère Bonnabel, Francis Bach

# The limited-memory recursive variational Gaussian approximation (L-RVGA)

Marc Lambert
DGA/CATOD, Centre d'Analyse Technico-Opérationelle de Défense
& INRIA - Ecole Normale Supérieure - PSL Research university
marc.lambert@inria.fr

Silvère Bonnabel
ISEA, Université de la Nouvelle-Calédonie
& MINES ParisTech, PSL University, Center for robotics
silvere.bonnabel@mines-paristech.fr

Francis Bach
INRIA - Ecole Normale Supérieure - PSL Research university
francis.bach@inria.fr

**Abstract**

We consider the problem of computing a Gaussian approximation to the posterior distribution of a parameter given a large number $N$ of observations and a Gaussian prior, when the dimension of the parameter $d$ is also large. To address this problem we build on a recently introduced recursive algorithm for variational Gaussian approximation of the posterior, called recursive variational Gaussian approximation (RVGA), which is a single pass algorithm, free of parameter tuning. In this paper, we consider the case where the parameter dimension $d$ is high, and we propose a novel version of RVGA that scales linearly in the dimension $d$ (as well as in the number of observations $N$), and which only requires linear storage capacity in $d$. This is afforded by the use of a novel recursive expectation maximization (EM) algorithm applied for factor analysis introduced herein, to approximate at each step the covariance matrix of the Gaussian distribution conveying the uncertainty in the parameter. The approach is successfully illustrated on the problems of high dimensional least-squares and logistic regression, and generalized to a large class of nonlinear models.

## 1 Introduction

In machine learning and statistics, Bayesian inference aims at estimating the full posterior distribution of a parameter of interest $\theta$, to better track the uncertainty of the learning process. Exact Bayesian inference is generally not tractable, and approximations are required. Variational approximation (Hinton and van Camp, 1993, Jordan et al., 1998) consists in rewriting the estimation problem as an approximate optimization problem over a restricted class of posterior distributions, such as Gaussian distributions, as advocated in the present paper.

More precisely, assume we want to approximate the posterior distribution $p(\theta|Y_N)$ of a Bayesian parameter $\theta$ given $N$ observations $Y_N = y_1, \ldots, y_N$. The variational Gaussian approximation (Barber and Bishop, 1998b, Opper and Archambeau, 2009, Challis and Barber, 2013) consists in minimizing the Kullback-Leibler divergence between this unknown posterior and a restricted class of

parameterized distributions, namely a Gaussian distribution $q(\theta|\mu, P) \sim \mathcal{N}(\theta|\mu, P)$, leading to the following optimization problem:

$$\min_{\mu,P} \quad KL\left(q(\theta|\mu, P)\|p(\theta|Y_N)\right) = \int q(\theta|\mu, P) \log \frac{q(\theta|\mu, P)}{p(\theta|Y_N)} d\theta. \tag{1}$$

In the recursive variational approach, the observations at each time step consist only of the last sample $y_t$, and the previous Gaussian estimate $q_{t-1}$ serves as a prior that sums up information obtained so far, so that one builds on the approximation $p(\theta|y_t) \propto p(y_t|\theta)q_{t-1}(\theta)$. This leads to a recursive variational approximation problem that writes at each step as follows:

$$q_t(\theta) = \arg\min_{\mu,P} \quad KL\left(\mathcal{N}(\theta|\mu, P)\|c_t p(y_t|\theta)q_{t-1}(\theta)\right), \tag{2}$$

where $c_t$ is a normalization constant which disappears when we consider the derivatives with respect to the variational parameters. It can be shown that the necessary stationary conditions for (2) yield the following updates, which are implicit—see (Lambert et al., 2021):

$$\mu_t = \mu_{t-1} + P_{t-1}\mathbb{E}_{\theta\sim\mathcal{N}(\mu_t,P_t)}[\nabla_\theta \log p(y_t|\theta)] \tag{3}$$

$$P_t^{-1} = P_{t-1}^{-1} - \mathbb{E}_{\theta\sim\mathcal{N}(\mu_t,P_t)}[\nabla^2_\theta \log p(y_t|\theta)]. \tag{4}$$

We recognize a second order online algorithm doing an adaptive gradient descent on the stochastic loss function $\ell_t(\theta) = -\log p(y_t|\theta, x_t)$. Although this algorithm seamlessly scales with the number of observations $N$, it has quadratic cost in the parameter dimension $d$, hindering its use in contexts involving high dimensions. In this paper, we consider a factor analysis (FA) structure to approximate the precision matrix $P^{-1}$ at each time step with a "low rank + diagonal" matrix, that is, $P^{-1} \approx WW^T + \Psi$ where $W \in \mathcal{M}(d \times p)$ is a rank $p$ matrix whith $p \ll d$ and $\Psi \in \mathcal{M}_d(\mathbb{R})$ is a diagonal matrix. This choice of decomposition allows for $d \times d$ matrix manipulation, but in a limited-memory fashion, in that we need to store a much reduced number of parameters, namely $d + d \times p$.

We reformulate the considered factorization in a recursive way such that the implicit equation (4) becomes:

$$W_{t-1}W_{t-1}^T + \Psi_{t-1} - \mathbb{E}_{\theta\sim\mathcal{N}(\mu_t,(W_tW_t^T+\Psi_t)^{-1})}[\nabla^2_\theta \log p(y_t|\theta)] \underset{\text{FA}}{\approx} W_tW_t^T + \Psi_t. \tag{5}$$

In the particular case of a linear regression model $y_t = x_t^T\theta + w_t$ with output noise $w_t \sim \mathcal{N}(0, 1)$ and with $x_t$ the input associated to the observation $y_t$, this recursive update becomes explicit:

$$W_{t-1}W_{t-1}^T + \Psi_{t-1} + x_tx_t^T \underset{\text{FA}}{\approx} W_tW_t^T + \Psi_t. \tag{6}$$

We recognize a classical factor analysis problem for the scaled empirical covariance of the inputs $\sum_{t=1}^N x_tx_t^T$, but done in a recursive way.

We show that the variational parameters $W_t$ and $\Psi_t$ can be computed through a recursive variant of the EM algorithm which is free of step tuning and offers a good approximation of the precision matrix after only one pass through the data.

We then generalize this recursive EM approach in the nonlinear case where we approximate the Hessian term in (5) with an explicit outer product and the expectation through sampling. To this aim we propose a novel sampling method to sample from $\mathcal{N}(\mu, (WW^T + \Psi)^{-1})$ without storing a $d \times d$ matrix, resulting in updates which scale linearly with $d$ in terms of both computation and storage costs.

The paper is organized as follows: in Section 2 we discuss how our approach differs from several related studies on large-scale variational inference. In Section 3, we model our problem as a two-stage variational approximation scheme and show how the second stage can be recast as an expectation-maximization problem. We then introduce in the linear case our new recursive EM algorithm with computation and storage cost being linear in $d$. In Section 4, we extend our approach to the nonlinear case using several approximations of the covariance update equation. We also introduce our novel method to sample from a Gaussian whose dispersion is encoded by a precision matrix structured with a factor analysis model.

In Section 5 our algorithm is evaluated on synthetic data for large scale matrix approximation and for linear and logistic regression.

## 2 Related work

Variational inference offers a way to approximate the distribution of latent parameters (Hinton and van Camp, 1993, Jordan et al., 1998) in Bayesian machine learning, and has garnered a lot of attention over the recent years for its ability to address big data problems that are otherwise generally out of reach. Although it may provide biased results, it converges much faster than the gold standard, Markov Chain Monte Carlo or MCMC (Andrieu et al., 2003), that is eventually accurate but can be slow. The model used in variational methods can be better specified using latent parameters (Minh-Ngoc et al., 2016, Linda and David, 2013).

To tackle large scale datasets, stochastic solvers are used to compute the unknown parameters leading to general frameworks such as the black box variational inference algorithm (Ranganath et al., 2014). In this article, we consider variational Gaussian approximation (VGA) where the variational parameters boil down to a mean vector and a covariance matrix. To tackle high dimensional problems in VGA, an approximation of the covariance matrix is needed. Challis and Barber (2013) have shown that, for a generalized linear model (GLM), the KL divergence is convex in the square root of the covariance matrix and have proposed different types of approximations for this square root matrix. Ong et al. (2018) considered a factor analysis structure, the factor analysis parameters being estimated using stochastic gradient descent. We consider also a factor analysis structure in this article but we do not optimize directly the parameters to avoid using a stochastic gradient descent which may be difficult to tune.

Mishkin et al. (2018) solved the variational problem in high dimension using an SVD to compute a factor analysis structure for the precision matrix at each step of a natural gradient descent. The two approximation schemes, Gaussian approximation and factor analysis approximation, are done sequentially. Our contribution extends this approach and differs from it in several ways. First, we do not consider a natural gradient descent algorithm on the variational parameters but we use instead the RVGA scheme which requires no step tuning (Lambert et al., 2021). Second, we propose a parameter-free recursive EM algorithm to compute the factor analysis precision matrix approximation which relaxes the need to resort to SVD, which can prove difficult in high dimension, and is well suited to accurately approximate the correlation between the low rank part and the diagonal part of the factor analysis structure. Moreover, we show that our approach has a nice interpretation as a two-stage variational inference that performs two "I-projections" (Csiszár and Shields, 2004) at each step: the first one on the space of Gaussian distributions, the second one on the space of Gaussian distributions with a structured precision matrix constrained to be "diagonal + low-rank".

Beyond variational inference, our work is also related to second order stochastic optimization where the Hessian is approximated online to scale the memory like in Adagrad (Duchi et al., 2011)

or TONGA Roux et al. (2008). Interesting connections may be done also with large scale Kalman filtering like the SEEK filter (Pham et al., 1998) which approximates the covariance matrix with an SVD decomposition or the the ensemble filter (Evensen, 1994) which approximates it with sampling.

# 3 Two-stage variational approximation

To introduce a variational loss for factor analysis, we recast the original recursive problem (2) as a two-stage variational problem as follows:

**Two-stage variational approximation**

$$q_t^*(\theta) = \mathcal{N}(\theta|\mu_t^*, P_t^*) = \arg\min_{\mu,P} \quad KL(\mathcal{N}(\theta|\mu, P)\|c_t p(y_t|\theta) q_{t-1}(\theta)) \tag{7}$$

$$q_t(\theta) = \mathcal{N}(\theta|\mu_t^*, \tilde{P}_t) = \arg\min_{W,\Psi} \quad KL(\mathcal{N}(\theta|\mu_t^*, (WW^T + \Psi)^{-1})\|q_t^*(\theta)). \tag{8}$$

The first variational problem is the projection of the posterior distribution onto the space of Gaussian distributions whereas the second variational problem is the projection of a Gaussian onto the space of structured Gaussian having a low rank plus diagonal structure for the precision matrix. This factorization aims to exploit the parsimony of the covariance matrix to limit the memory requirement.

The first projection was the object of our previous work (Lambert et al., 2021) and leads to the RVGA update giving the optimal solution for the mean (3) and covariance (4), whereas the second one can be addressed through an expectation-maximization algorithm as we show now.

## 3.1 Low rank + diagonal approximation via EM

We first show that the divergence in (8) may be algebraically related to a maximum likelihood problem. Indeed, since the means are the same at both sides of the KL divergence, we may ignore them and switch sides once a matrix inversion has been done as follows:

$$KL(\mathcal{N}(\theta|\mu_t^*, (WW^T + \Psi)^{-1})\|\mathcal{N}(\theta|\mu_t^*, P_t^*)) \tag{9}$$

$$= KL(\mathcal{N}(\theta|0, P_t^{*-1})\|\mathcal{N}(\theta|0, WW^T + \Psi)) \tag{10}$$

$$= \frac{1}{2}\mathrm{Tr}((WW^T + \Psi)^{-1} P_t^{*-1}) + \frac{1}{2}\log\det(WW^T + \Psi) - \frac{1}{2}\log\det(P^{*-1}) - \frac{d}{2}. \tag{11}$$

Let us consider $K$ samples $v_1, \ldots, v_K$ supposed to be centered and such that their empirical covariance is $S_K = \frac{1}{K}\sum_{i=1}^K v_i v_i^T = P_t^{*-1}$. The log-likelihood on these samples is:

$$\max_{W,\Psi} \quad \log\mathcal{N}(v_1, \ldots, v_K|0, WW^T + \Psi) \tag{12}$$

$$= \max_{W,\Psi} \quad -\frac{K}{2}\mathrm{Tr}((WW^T + \Psi)^{-1}\frac{1}{K}\sum_{i=1}^K v_i v_i^T) - \frac{K}{2}\log\det(WW^T + \Psi) - \frac{K}{2} d\log(2\pi), \tag{13}$$

where we have used the relation $\sum_{i=1}^K v_i^T (WW^T + \Psi)^{-1} v_i = \mathrm{Tr}((WW^T + \Psi)^{-1}\sum_{i=1}^K v_i v_i^T)$. We see that minimizing the divergence (11) is equivalent to maximizing the log-likelihood (13).

The variational parameters can be obtained by zeroing the derivative of the maximum likelihood (ML) and using a singular value decomposition to find the solution. We will rather consider an expectation-maximization (EM) algorithm (Dempster et al., 1977) which better scales to high-dimensional problems and which is guaranteed to increase the (total) likelihood at each step (see

Donald and Dorothy, 1982). It can be shown that the EM approach computes implicitly a singular value decomposition to find the latent factors but in a memory-efficient way. The connexion between the fixed-point ML, the fixed-point EM approach and the associated eigenvalue algorithms is discussed in more details in Appendix B.

To apply the EM algorithm we need to introduce a latent variable $z$ such that our samples take the form $v_i = Wz_i + \varepsilon_i$ where $\varepsilon_i \sim \mathcal{N}(0, \Psi)$.

At the expectation step (E-step), the latent variables $z_i$ are estimated conditionally on the observation and the current variational parameters $W$ and $\Psi$ using the conditioning formula:

$$p(z_i|v_i, W, \Psi) = \mathcal{N}(M^{-1}W^T\Psi^{-1}v_i, M^{-1}) \text{ where } M = \mathbb{I}_p + W^T\Psi^{-1}W. \tag{14}$$

At the maximization step (M-step), the variational parameters are adjusted to maximize the expected total likelihood defined by:

$$\mathbb{E}[\log p(v_1, z_1, \ldots, v_K, z_K|W, \Psi)] = \mathbb{E}[\sum_{i=1}^{K} \log p(v_i|z_i, W, \Psi) + \sum_{i=1}^{K} \log p(z_i)] \tag{15}$$

$$= \mathbb{E}[-\frac{K}{2}(v_i - Wz_i)^T\Psi^{-1}(v_i - Wz_i) - \frac{K}{2}\log\det\Psi - \frac{K}{2}\log(2\pi) + \sum_{i=1}^{K}\log p(z_i)], \tag{16}$$

where the expectations are taken over the conditional distribution $z_i \sim p(z_i|v_i, W, \Psi)$, where parameters $W, \Psi$ are there fixed to their current value. After some calculations recapped in Appendix A.1, the EM algorithm for the factor analysis problem yields the following updates:

**E-Step:**

$$\begin{aligned}\mathbb{E}[z_i|v_i] &= M^{-1}W^T\Psi^{-1}v_i \\ \mathbb{E}[z_iz_i^T|v_i] &= M^{-1} + \mathbb{E}[z_i|v_i]\mathbb{E}[z_i|v_i]^T.\end{aligned} \tag{17}$$

**M-Step:**

$$\begin{aligned}W^{(n)} &= \sum_{i=1}^{K} v_i\mathbb{E}[z_i|v_i]^T(\sum_{i=1}^{K}\mathbb{E}[z_iz_i^T|v_i])^{-1} \\ \Psi^{(n)} &= \operatorname{diag}\Big(\frac{1}{K}\sum_{i=1}^{K} v_iv_i^T - W^{(n)}\frac{1}{K}\sum_{t=1}^{K}\mathbb{E}[z_i|v_i]v_i^T\Big),\end{aligned} \tag{18}$$

where the $^{(n)}$ stands for "new".

The expectation and maximization steps can be computed through a single update leading to the following fixed-point scheme (see Appendix A.1 for further details):

$$W^{(n)} = S_K\Psi^{-1}W(\mathbb{I}_p + M^{-1}W^T\Psi^{-1}S_K\Psi^{-1}W)^{-1}, \tag{19}$$

$$\text{where } M = \mathbb{I}_p + W^T\Psi^{-1}W$$

$$\begin{aligned}\Psi^{(n)} &= \operatorname{diag}(S_K - W^{(n)}M^{-1}W^T\Psi^{-1}S_K) \\ W &= W^{(n)}, \quad \Psi = \Psi^{(n)}.\end{aligned} \tag{20}$$

This update no longer depends on the samples $v_i$ nor on the latent parameters $z_i$ explicitly. It only makes use of $S_K = \frac{1}{K}\sum_{i=1}^{K} v_iv_i^T = P_t^{*-1}$. The matrix $P_t^{*-1}$ is the output of the R-VGA update

(4) and depends on the previous estimate ${P_{t-1}}^{-1}$, supposed already in a factor analysis form, such that we have the relation:

$$P_t^{*-1} = W_{t-1}W_{t-1}^T + \Psi_{t-1} - \mathbb{E}_{\theta\sim\mathcal{N}(\mu_t,P_t)}[\nabla_\theta^2 \log p(y_t|\theta)]. \tag{21}$$

Substituting this expression into our fixed point EM algorithm, we obtain a closed form for the second-stage variational update. But this update involves the $d \times d$ matrix $\nabla_\theta^2 \log p(y_t|\theta)$, which does not fit in memory in large scale problems. Moreover, this matrix is not well defined since it depends on an expectation under unknown parameters $\mathbb{E}_{\theta\sim\mathcal{N}(\mu_t,P_t)}$. The goal of Section 4 is to approximate this matrix to obtain a tractable and limited memory algorithm. Before that, we consider an interesting particular case where the model is linear. We show that in this case, the recursive equations we have just derived allow for an actual novel EM algorithm, which is both online and memory-efficient.

### 3.2 The linear case: a recursive EM algorithm for factor analysis

Let us consider for now the simpler case of a linear model $y_t = x_t^T\theta + w_t$, where $w_t \sim \mathcal{N}(0, \sigma_w^2)$. The variance $\sigma_w^2$ is a scaling factor we will set equal to 1. The RVGA updates (3)-(4) are then equivalent to the following explicit updates in the linear case, see Lambert et al. 2021:

$$\mu_t = \mu_{t-1} + P_t x_t(y_t - x_t^T\mu_{t-1}) \tag{22}$$

$$P_t^{-1} = P_{t-1}^{-1} + x_t x_t^T. \tag{23}$$

If we apply our two-stage variational approximation in this linear case, the first stage approximation (7) is exactly solved by the update above and the second stage approximation (8) addresses limited-memory requirements. Using the factor analysis approximation in a recursive way, we obtain:

$$\mu_t = \mu_{t-1} + (W_tW_t^T + \Psi_t)^{-1}x_t(y_t - x_t^T\mu_{t-1}), \tag{24}$$

$$W_tW_t^T + \Psi_t \underset{FA}{\approx} W_{t-1}W_{t-1}^T + \Psi_{t-1} + x_tx_t^T. \tag{25}$$

Using the Woodbury formula $(W_tW_t^T + \Psi_t)^{-1}x_t = \Psi_t^{-1}(x_t - W_tM_t^{-1}(W_t^T\Psi_t^{-1}x_t))$ with $M_t = \mathbb{I}_p + W_t^T\Psi_t^{-1}W_t$, (24) can be rewritten in a limited-memory fashion involving only operations linear in $d$:

$$(24) \Leftrightarrow \mu_t = \mu_{t-1} + \Psi_t^{-1}(x_t - W_t(\mathbb{I}_p + W_t^T\Psi_t^{-1}W_t)^{-1}(W_t^T\Psi_t^{-1}x_t))(y_t - x_t^T\mu_{t-1}). \tag{26}$$

To address (25), we can use our fixed-point equation replacing the matrix $S_K$ in (20) by $W_{t-1}W_{t-1}^T + \Psi_{t-1} + x_tx_t^T$. This defines a recursive EM algorithm, namely Algorithm 1, which consists in successively performing a few cycles on the fixed-point equation (20), where we develop and rearrange the terms to avoid any operations that require storing or multiplying $d \times d$ matrices. The Algorithm 1 is given in a more general setting where we solve a factor analysis approximation of the form:

$$W_tW_t^T + \Psi_t \underset{FA}{\approx} \alpha_t(W_{t-1}W_{t-1}^T + \Psi_{t-1}) + \beta_t x_tx_t^T, \tag{27}$$

with $\alpha_t, \beta_t$ scalar constants being equal to 1 in this section. The reason for these parameters is that it will prove useful to come back to this approximation with different coefficients when we turn to the nonlinear case.

Using the recursive EM in this way gives a factor analysis decomposition of the precision matrix $P_t^{*-1}$, which corresponds to an increasing amount of information $P_0^{-1} + \sum_{i=1}^{t} x_i x_i^T$ as more inputs are processed, where $P_0$ corresponds to the covariance of the prior.

This prior matrix $P_0$ can not be stored in memory and the algorithm 1 contains an initialization procedure which computes $W_0$ and $\Psi_0$ such that $W_0 W_0^T + \Psi_0 \approx P_0^{-1}$, see Section 5.1.2 for more details, in the numerical experiments section.

**Algorithm 1:** Recursive expectation-maximization algorithm for solving factor analysis approximation (27)

**Result:** $W, \Psi$
Given $N$ inputs $x_1, \dots, x_N$ in high dimension $d$;
Given a latent dimension $p \ll d$;
Given a prior covariance $P_0$;
**-Initialization (consistent with $P_0$, see 5.1.2)-**
Initialize $W \in \mathcal{M}(d \times p)$ and $\Psi \in \mathcal{M}_d(\mathbb{R})$ diagonal:
$\Psi = \Psi_0$ ;
$W = W_0$;
**-Update-**
**for** $t \leftarrow 1$ **to** $N$ **do**
  Access an input $x_t$;
  **for** $k \leftarrow 1$ **to** *nbInnerLoop* **do**
    $M = \mathbb{I}_p + W^T \Psi^{-1} W$;
    $V = \beta_t x_t (x_t^T \Psi^{-1} W) + \alpha_t (W_{t-1} (W_{t-1}^T \Psi^{-1} W) + \Psi_{t-1} \Psi^{-1} W)$ ;
    $W^{(n)} = V (\mathbb{I}_p + M^{-1} W^T \Psi^{-1} V)^{-1}$;
    $\Psi^{(n)} = \beta_t x_t * x_t + \alpha_t (W_{t-1} * W_{t-1} + \Psi_{t-1}) - W^{(n)} M^{-1} * V$ ;
    $W = W^{(n)}$ ;
    $\Psi = \Psi^{(n)}$ ;
  **end**
  $W_{t-1} = W$;
  $\Psi_{t-1} = \Psi$;
**end**
/* From 1 to 3 inner loops (*nbInnerLoop*) may be sufficient to make the algorithm converge. The initialization explain in Section 5.1.2 compute $W_0$ and $\Psi_0$ such that $W_0 W_0^T + \Psi_0 \approx P_0^{-1}$. The operation "$*$" which appears at the last line aim to compute $X * Y = \mathrm{diag}(XY^T)$ in a memory efficient way. It is applied on two matrices $X$ and $Y$ of same size $d \times p$. If $p = 1$, this operator match the component wise operator $\odot$ : $X * Y = x \odot y$. If $p > 1$, it writes $X * Y = \sum_{i=1}^{p} x[., i] \odot y[., i]$. The diagonal matrix $\Psi$ is stored and used as a vector: any multiplication of the diagonal matrix with a vector may be computed faster as an element-wise product: $\Psi^{-1} W = W / \psi$ where $\psi = \mathrm{diag}(\Psi)$ and $/\psi$ operates on columns of $W$ and $W^T \Psi^{-1}$ gives as well $W^T / \psi^T$ where $/\psi^T$ operates on the rows of $W^T$. */

# 4 The limited memory recursive variational Gaussian approximation (L-RVGA)

In this Section, we consider the extension of our algorithm when the observations depend non-linearly on the latent parameter $\theta$. We first extend the previous linear case to generalized linear models in Section 4.1. Section 4.2 considers the wholly general case, which necessitates additional approximations.

## 4.1 L-RVGA for generalized linear models

The latter approach may be extended whenever the observation $y_t$ follows an exponential family distribution, which may be advantageous in the context of classification problems. An exponential family distribution (Pitcher, 1979) takes the form:

$$p(y_t) = c(y_t) \exp(\eta^T y_t - F(\eta)), \tag{28}$$

where $F$ is the log partition function and $c(y)$ is a normalization function. It can represent a large family of distributions like the Gaussian, multinomial or Dirichlet distribution. The natural parameter $\eta$ is related to the expectation parameter $m = \mathbb{E}[y]$ through the link function $g$ such that $\eta = g(m)$. In the machine learning framework, the natural parameter is also related to the input $x$ and the hidden latent parameter $\theta$ through a linear function $\eta = \theta^T x$, this model is called the generalized linear model with a canonical natural parameter. The advantage of this model is that the Hessian term takes the form of a (generalized) outer product $-\nabla^2_\theta \log p(y_t|\theta) = x_t \nabla^2_\eta F(\eta(\theta)) x_t^T$. In this case the RVGA updates (3)-(4) become:

$$P_t^{-1} = P_{t-1}^{-1} + \gamma_t x_t x_t^T \tag{29}$$

$$\mu_t = \mu_{t-1} + \xi_t P_{t-1} x_t, \tag{30}$$

where we have introduced the scalar weighting parameter $\gamma_t$ and the scalar error $\xi_t$ which depend on $\theta$ as follows

$$\gamma_t = \mathbb{E}_{\theta \sim \mathcal{N}(\mu_t, P_t)}[\mathrm{Cov}(y_t|\theta)], \tag{31}$$

$$\xi_t = y_t - \mathbb{E}_{\theta \sim \mathcal{N}(\mu_t, P_t)}[m(y|\theta)], \tag{32}$$

and where we have used the properties of exponential family $\nabla_\eta F(\eta(\theta)) = m(y|\theta)$ and $\nabla^2_\eta F(\eta(\theta)) = \mathrm{Cov}(y|\theta)$, see Lambert et al. (2021). The factorized approximation of this update can be performed through Algorithm 1, since the factor analysis approximation $W_{t-1}W_{t-1}^T + \Psi_{t-1} + \gamma_t x_t x_t^T \underset{\text{FA}}{\approx} W_t W_t^T + \Psi_t$ of (29) fits into the problem (27), letting $\alpha_t = 1, \beta_t = \gamma_t$. The computation of the scalar terms $\gamma_t$ and $\xi_t$ is nontrivial, and requires resorting to approximations, though. Those approximations may rely on the tools developed in the next subsection, devoted to the general case. However, in the case of logistic regression with $m(y|\theta) = \frac{1}{1+\exp(-\theta^T x)}$ and $\mathrm{Cov}(y|\theta) = m(y|\theta)(1 - m(y|\theta))$ these parameters were shown in our prior work on RVGA Lambert et al. (2021) to be easily obtainable numerically and will serve as a baseline for our experimentations in Section 5.3.

## 4.2 Limited memory RVGA (L-RVGA) : nonlinear model

As before, we suppose that the observation $y_t$ follows the exponential family distribution (28) but with a nonlinear dependence of the form $\eta = h(\theta, x)$ with $h$ is a nonlinear function. In our setting,

we need to approximate the matrix $P_t^{-1}$ recursively, as follows

$$P_t^{-1} = W_{t-1}W_{t-1}^T + \Psi_{t-1} - \mathbb{E}_{\theta\sim\mathcal{N}(\mu_t,P_t)}[\nabla_\theta^2 \log p(y_t|\theta, x_t)] \tag{33}$$
$$\approx W_{t-1}W_{t-1}^T + \Psi_{t-1} + X_t X_t^T \tag{34}$$
$$\underset{\text{FA}}{\approx} W_t W_t^T + \Psi_t,$$

where $X_t$ is a rectangular matrix supposed to fit into memory which will be defined more precisely in Section 4.2.4. Using this approximation, we can run the recursive EM Algorithm 1 in a memory-efficient way, replacing the input $x_t$ by the matrix $X_t$. The mean is computed as before:

$$\mu_t = \mu_{t-1} - (W_{t-1}W_{t-1}^T - \Psi_{t-1})^{-1}\mathbb{E}_{\theta\sim\mathcal{N}(\mu_t,P_t)}[\nabla_\theta \log p(y_t|\theta, x_t)], \tag{35}$$

where we can use the Woodbury formula again to compute efficiently the weighting term $(W_{t-1}W_{t-1}^T - \Psi_{t-1})^{-1}$.

A practical implementation of these updates raises the following difficulties:

1. Computing $P_t^{-1}$ and $\mu_t$ in (33)-(35) involves computing an expectation over a distribution parameterized by $\mu_t$ and $P_t$, i.e., the scheme is implicit.

2. As no closed form is generally available for the computation of the expectations $\mathbb{E}_{\theta\sim\mathcal{N}(\mu,P)}$, one may resort to Monte-Carlo sampling. However, in this paper we maintain a limited memory approximation of the *inverse* of the covariance matrix, that is, $P^{-1} = WW^T + \Psi$ and we need to sample from a Gaussian $\mathcal{N}(\mu, P)$ without storing or inverting a $d \times d$ matrix.

3. The Hessian matrix $\nabla_\theta^2 \log p(y_t|\theta, x_t)$ (or its averaged value) must be computable and stored with linear cost in the dimension $d$.

In the remainder of the section, we address all these points.

### 4.2.1 Using extra-gradients for the implicit scheme

The more direct way to address Point 1 above is to open the loop and to replace the expectations under $\mathcal{N}(\mu_t, P_t)$ with expectations under $\mathcal{N}(\mu_{t-1}, P_{t-1})$. However experiments we conducted showed that this naive scheme can lead to instability. The importance of managing the implicit scheme was the object of our previous work (Lambert et al., 2021), where we have developed closed-form formulas to solve the implicit scheme in the linear and logistic regression case. In the general case, one may resort to extra-gradient, i.e., to update first the covariance and then the mean, and to iterate twice:

**Iterated RVGA**

$$\begin{aligned}
\hat{\mathbf{P}}_\mathbf{t}^{-1} &= P_{t-1}^{-1} - \mathbb{E}_{\theta\sim\mathcal{N}(\mu_{t-1},P_{t-1})}[\nabla_\theta^2 \log p(y_t|\theta)] \\
\hat{\mu}_\mathbf{t} &= \mu_{t-1} + \hat{\mathbf{P}}_\mathbf{t}\mathbb{E}_{\theta\sim\mathcal{N}(\mu_{t-1},P_{t-1})}[\nabla_\theta \log p(y_t|\theta)] \\
\mathbf{P}_\mathbf{t}^{-1} &= P_{t-1}^{-1} - \mathbb{E}_{\theta\sim\mathcal{N}(\hat{\mu}_\mathbf{t},\hat{\mathbf{P}}_\mathbf{t})}[\nabla_\theta^2 \log p(y_t|\theta)] \\
\mu_t &= \mu_{t-1} = \mathbf{P}_\mathbf{t}\mathbb{E}_{\theta\sim\mathcal{N}(\hat{\mu}_\mathbf{t},\hat{\mathbf{P}}_\mathbf{t})}[\nabla_\theta \log p(y_t|\theta)].
\end{aligned} \tag{36}$$

It turns out that this iterated scheme is equivalent to the "Mirror Prox" algorithm (Nemirovski, 2005), a.k.a., extra-gradient, with a unit step size and applied to the function $f(\mu, P) = \mathbb{E}_{\theta\sim\mathcal{N}(\mu,P)}[\log p(y|\theta)]$. Mirror Prox is known to help convergence on a large set of problems (convex optimization, variational inequalities) and we have experimentally observed that this iterated scheme reduces significantly the bias of our estimator (see Appendix E for further details).

However, when we combine extra-gradient with factor analysis, the extra covariance update can make the Mirror Prox scheme unstable. We have observed it is then preferable to skip the extra covariance update, that is, the third line of the iterated scheme above (see Appendix E). This choice has been made in Section 5.4 dedicated to experiments in the general case.

#### 4.2.2 Gaussian sampling from the precision matrix

To address Point 2 above, we need to sample efficiently from $\mathbb{E}_{\theta\sim\mathcal{N}(\mu,(WW^T+\Psi)^{-1})}$ without storing a $d\times d$ matrix. This problem was already addressed by Mishkin et al. (2018) and Ambikasaran et al. (2014) using a square-root form and two Cholesky decompositions on the latent space. However, we propose here a faster method inspired by the ensemble Kalman filter (Evensen, 1994) which does not require computing any Cholesky decomposition.

**Proposition 1.** *Let us define the quantities:* $M = \mathbb{I}_p + W^T\Psi^{-1}W$ *and* $L = \Psi^{-1}WM^{-1}$. *Draw* $x\sim\mathcal{N}(0,\Psi^{-1})$ *and* $\epsilon\sim\mathcal{N}(0,\mathbb{I}_p)$ *independently, and define*

$$x^+ = x + L(\epsilon - LW^Tx) = (I - LW^T)x + L\epsilon.$$

*We have then* $x^+\sim\mathcal{N}(0,P)$, *with*

$$P = (WW^T+\Psi)^{-1} = \Psi^{-1} - \Psi^{-1}W(\mathbb{I}_p + W^T\Psi^{-1}W)^{-1}W^T\Psi^{-1}.$$

*Proof.* We have obviously $E(x^+)=0$. Moreover using the independence of the variables

$$\begin{aligned} E\left(x^+(x^+)^T\right) &= (I-LW^T)\Psi^{-1}(I-LW^T)^T + L\mathbb{I}_pL^T && (37)\\ &= \Psi^{-1} - LW^T\Psi^{-1} - \Psi^{-1}WL^T + L[W^T\Psi^{-1}W+\mathbb{I}_p]L^T && (38)\\ &= \Psi^{-1} - \Psi^{-1}W(\mathbb{I}_p + W^T\Psi^{-1}W)^{-1}W^T\Psi^{-1}, && (39)\end{aligned}$$

where we used that the three rightmost terms of (38) are all equal to $\pm\Psi^{-1}W(\mathrm{I}_p+W^T\Psi^{-1}W)^{-1}W^T\Psi^{-1}$. □

This suggests that, starting from a decomposition of the form $WW^T+\Psi$ of the precision matrix $P^{-1}$, we know how to sample from a law $\mathcal{N}(0,P)$. It suffices to draw $x_1,\dots,x_K$ from $\mathcal{N}(0,\Psi^{-1})$ and $\epsilon_1,\dots,\epsilon_K$ from $\mathcal{N}(0,\mathbb{I}_p)$, and to let $x_i^+ = x_i + L(\epsilon_i - LW^Tx_i) = (I-LW^T)x_i + L\epsilon_i$, for $1\le i\le K$. All operations involved are linear in $d$, so that drawing $K$ samples is of order $O(Kd)$. As soon as $K$ is kept moderate with respect to $d$, this may be considered as having linear complexity in $d$.

#### 4.2.3 Dealing with the $d\times d$ Hessian matrix

In this paragraph we address Point 3 of the list above, where we propose an approximation of the Hessian based on an outer product. The Gauss-Newton approximation, which consists in approximating the Hessian with the outer product of the gradients, called also empirical Fisher, does not capture well second-order information (see for instance Kunstner et al. 2019). As we have supposed the probability distribution belongs to an exponential family, we should rather consider the Generalized Gauss-Newton (GGN) approximation (Martens, 2014) which better approximates the local curvature (Kunstner et al., 2019). The GGN approximation exploits the structure of the exponential family $p(y_t|\eta = h(\theta,x))$ as a composition of functions involving the natural parameter $\eta$ and the nonlinear model $h$. If the Hessian of $-\log p$ with respect to $\theta$ can be complicated, the Hessian

with respect to $\eta$ is simply $\text{Cov}(y|\theta)$. We can use this property through the chain rule to generalize the Gauss-Newton approximation. Our expected Hessian term can be written as follows:

$$-\mathbb{E}_\theta[\nabla^2_\theta \log p(y_t|\theta)] \approx \mathbb{E}_\theta[\frac{\partial h}{\partial \theta}\text{Cov}(y|\theta)\frac{\partial h}{\partial \theta}^T] = \mathbb{E}_\theta[\mathbb{F}(\theta)], \tag{40}$$

where $\mathbb{F}(\theta) = \mathbb{E}_{y\sim p(y|\theta)}[\nabla^2_\theta \log p(y|\theta)]$ is the Fisher matrix. Under the GGN approximation, the covariance update no longer depends on the labels $y_t$ and only depends on the inputs $x_t$, as in the linear case. The derivation is detailed in Appendix D. We can now combine this approximation with the approximation of the expectation and the implicit scheme to implement our final update.

#### 4.2.4 Final algorithm

We now define the form of matrix $X_t$ involved in the outer product in (34). Assuming that we make the scheme explicit using extra-gradients (as described in Section 4.2.1) and that we generate $K$ samples to approximate the expectation with Ensemble sampling described in Section 4.2.2, the GGN approximation becomes:

$$P_t^{-1} \underset{\text{GGN}}{\approx} W_{t-1}W_{t-1}^T + \Psi_{t-1} + \mathbb{E}_{\theta\sim\mathcal{N}(\mu_t,P_t)}[\frac{\partial h}{\partial \theta}\text{Cov}(y|\theta)\frac{\partial h}{\partial \theta}^T] \tag{41}$$

$$\underset{\text{Extragrad}}{\approx} W_{t-1}W_{t-1}^T + \Psi_{t-1} + \mathbb{E}_{\theta\sim\mathcal{N}(\mu_{t-1},(W_{t-1}W_{t-1}^T+\Psi_{t-1})^{-1})}[\frac{\partial h}{\partial \theta}\text{Cov}(y|\theta)\frac{\partial h}{\partial \theta}^T] \tag{42}$$

$$\underset{\text{Sampling}}{\approx} W_{t-1}W_{t-1}^T + \Psi_{t-1} + \frac{1}{K}\sum_{i=1}^K \frac{\partial h}{\partial \theta}(\theta_i)\text{Cov}(y|\theta_i)^{1/2}\text{Cov}(y|\theta_i)^{1/2}\frac{\partial h}{\partial \theta}^T(\theta_i) \tag{43}$$

$$= W_{t-1}W_{t-1}^T + \Psi_{t-1} + \frac{1}{K}\sum_{i=1}^K c_i c_i^T \quad \text{where} \quad c_i = \frac{\partial h}{\partial \theta}(\theta_i)\text{Cov}(y|\theta_i)^{1/2} \tag{44}$$

$$= W_{t-1}W_{t-1}^T + \Psi_{t-1} + X_t X_t^T \quad \text{where} \quad X_t = \frac{1}{\sqrt{K}}\left(c_1 \cdots c_K\right) \tag{45}$$

$$\underset{\text{FA}}{\approx} W_t W_t^T + \Psi_t, \tag{46}$$

where the last line refers to the FA approximation developed in Section 3.2 applied to the mini-batch matrix $X_t$ of size $d \times K$. As long as $K \ll d$, the memory cost is kept linear in $d$.

## 5 Experiments

Experiments have been performed on synthetic data for different classes of problems, from simpler linear problems to more difficult nonlinear problems. The first class of problems addressed in Section 5.1.3 deals with the approximation of large scale covariance matrices that are too large to fit into memory. We show our approach requires far less memory than the batch EM algorithm which requires storing a $d \times d$ matrix in memory and competes with the online EM algorithm (Cappé and Moulines, 2009). These results are directly applied to linear regression in Section 5.2 where we propose computationally cheap updates for the linear Kalman filter in high dimensional spaces. In Section 5.3, we consider the logistic regression problem for which we can solve the recursive variational scheme in closed form at each step (Lambert et al., 2021). Finally, we turn to more general cases in Section 5.4 where we assess our approximation method for L-RVGA updates in the general case. The logistic regression problem, where we know how to compute the expectation analytically, offers a baseline for comparison purposes.

We observe the mirror-prox method combined with ensemble sampling allows for reaching the baseline results in terms of KL divergence. This is promising regarding the generalization of our algorithms to arbitrary nonlinear problems.

## 5.1 Limited-memory approximation of large scale covariance matrices

The first considered experimental problem is large scale covariance approximation. It is first shown to be amenable to the class of problems we have considered hitherto. Before turning to numerical experiments per se, we discuss also initialization of Algorithm 1.

### 5.1.1 Recursive factor analysis of the covariance matrix

The retained factorization $W_t W_t^T + \Psi_t$ of the precision matrix $P_t^{-1}$ associated to the Bayesian parameter $\theta_t$ is not conventional in factor analysis, which is usually applied to the empirical covariance matrix $S_t = \frac{1}{t}\sum_{i=1}^t x_i x_i^T$ of the inputs $x_t$. In the linear case, both can be related as follows $S_t := \frac{1}{t}P_t^{-1}$ for $t > 0$, and at $t = 0$ we let $S_0 = P_0^{-1}$. $S_t$ can be expressed in a recursive way as:

$$S_t = \frac{1}{t}P_t^{-1} = \frac{t-1}{t}\frac{P_{t-1}^{-1}}{t-1} + \frac{1}{t}x_t x_t^T = \frac{t-1}{t}S_{t-1} + \frac{1}{t}x_t x_t^T. \tag{47}$$

If we let $S_0$ be null, we obtain exactly $\frac{1}{t}\sum_{i=1}^t x_i x_i^T$, otherwise we obtain a regularized version of the empirical covariance. The corresponding recursive factor analysis form is:

$$S_t = \frac{t-1}{t}(W_{t-1}W_{t-1}^T + \Psi_{t-1}) + \frac{1}{t}x_t x_t^T, \tag{48}$$

which fits into the problem (27). It may thus be addressed through Algorithm 1. We thus recover a recursive EM analysis empirical covariance matrix factorization which is both online and memory-efficient. The online EM (Cappé and Moulines, 2009) is also a limited-memory version of the EM which gives state-of-the-art results when its learning parameters are finely tuned, but our approach is parameter-free and appears as natural in the context of factor analysis since it is derived directly from our two-stage variational approach. Moreover, experiments of Section 5.1.3 indicate it converges faster than online EM and gives new state-of-the-art results for online EM applied to factor analysis.

### 5.1.2 Initialization and prior information

Given a prior covariance $P_0$ on $\theta$, we want to initialize $W_0$ and $\Psi_0$ such that $W_0 W_0^T + \Psi_0 \approx P_0^{-1}$. Let's note $\sigma_0 = \sqrt{\frac{\mathrm{Tr}P_0}{d}}$ and approximate $P_0$ by $\sigma_0^2 \mathbb{I}_d$. This approximation is exact in the case of an isotropic covariance. We need now to choose $W_0$ and $\Psi_0$ such that $W_0 W_0^T + \Psi_0 \approx \frac{1}{\sigma_0^2}\mathbb{I}_d$. The simple choice $\Psi_0 = \frac{1}{\sigma_0^2}\mathbb{I}_d$ and $W_0 = 0_{d\times p}$ would make the algorithm run into problems as $W_0 = 0_{d\times p}$ is a stationary point of the fixed-point equation (20).

We use the following rule $\Psi_0 = \psi_0 \mathbb{I}_d$ where $\psi_0 > 0$ is a scalar and generate $W_0$ as a $d \times p$ matrix whose columns are the vectors $u_0^1, u_0^2, \ldots, u_0^p$ independently drawn from an isotropic Gaussian distribution in $\mathbb{R}^d$ and which have been normalized so that $\forall k : ||u_0^k|| = w_0$. We then let:

$$\psi_0 = (1-\varepsilon)\frac{1}{\sigma_0^2}, \qquad w_0 = \sqrt{\frac{\varepsilon d}{p}}\frac{1}{\sigma_0},$$

with $0 < \varepsilon \ll 1$ a small parameter. The rationale is that $W_0 W_0^T = \sum_{k=1}^p u_0^k u_0^{k^T}$ so that we have:

$$\mathrm{Tr}(W_0 W_0^T + \Psi_0) = \sum_{k=1}^p \mathrm{Tr}(u_0^k u_0^{k^T}) + \psi_0 \mathrm{Tr}\mathbb{I}_d = p w_0^2 + d\psi_0 = \frac{d}{\sigma_0^2} = \mathrm{Tr}P_0^{-1}.$$

**Remark 1.** *This initialization can be extended to the case where $P_0$ is a diagonal matrix $P_0 = \mathrm{Diag}(\sigma_1^2, \ldots, \sigma_d^2)$ where $\sigma_i^2$ represents now a variance on the $i^{th}$ coordinate.*

In the case where we want to approximate a large scale covariance matrix, we may guess the value of $S_0$ to initialize the procedure and make it more stable. Observing that $\mathrm{Tr}\frac{1}{N}\sum_{t=1}^N x_t x_t^T = \frac{1}{N}\sum_{t=1}^N ||x_t||^2$ we see that the trace of the unknown covariance must match the expectation of the square norm of inputs. This expectation may be estimated on a batch of size $M$. We use this property to compute $S_0$ as follows: get a batch formed by the first $M$ data inputs $x_1, \ldots, x_M$ , and let $S_0 = \frac{1}{\sigma_0^2}\mathbb{I}$ with $\sigma_0 = \sqrt{\frac{d}{\frac{1}{M}\sum_{t=1}^M ||x_t||^2}}$. This is equivalent to normalize the inputs in mean and set $\sigma_0 = \sqrt{d}$. This procedure is used in the following experiments in the Section 5.1.3.

### 5.1.3 Factorization of a synthetic large scale covariance matrix

In this section, we assess the limited-memory recursive EM algorithm on large scale empirical covariance matrices. We use Algorithm 1 with $\alpha_t = (t-1)/t$ and $\beta_t = 1/t$, see (48).

To assess the method, we generate the data from an actual low-rank covariance plus diagonal matrix $S = \mathrm{Diag}(\psi) + WW^T$ for which the factor analysis approximation can be exact. We construct such a matrix with random parameters where $W$ is of size $d \times p$ and $\psi$ is a vector of size $d$. We then generate $N$ samples from the zero-mean Gaussian distribution equipped with this covariance matrix $S$. To make the experiments more appealing, we have also assessed our algorithm on the NIPS Madelon dataset and the Breast Cancer real dataset. We have kept only the input data, using the LIBSVM library (Chang and Lin, 2011), and normalized them using the same process as for the synthetic dataset.

We run the batch EM algorithm on this set of $N$ samples and our limited memory EM algorithm recursively on each sample. The recursive EM generally converges to the result given by the batch version after only one pass through the data as shown in Figure 1. Our recursive EM is also compared to the online EM algorithm (Cappé and Moulines, 2009) which is also a limited memory algorithm but based on a stochastic root finding algorithm. The derivation of the online EM for factor analysis and the implementation detailed have been moved to Appendix C. While our method is parameter free it yields results being similar to the online EM on the synthetic dataset and better results on the real datasets as shown in upper plots of Figure 1.

The advantage of online or recursive EM versions for factor analysis is that they maintain a memory cost linear in $d$, which is an important feature in high dimension. To illustrate this feature, we performed large scale covariance matrix approximation in dimension 1 million. We see in Table 1 that the batch EM does not fit the memory of a standard laptop, whereas the proposed algorithm may be performed using a laptop.

## 5.2 Application to Bayesian linear regression

We apply in this section the previous covariance approximation to derive a limited memory version of the linear regression problem as described in Section 3.2. In the linear general case, we have an analytical form of the solution given by the linear Kalman filter with a static state which constitutes our baseline.

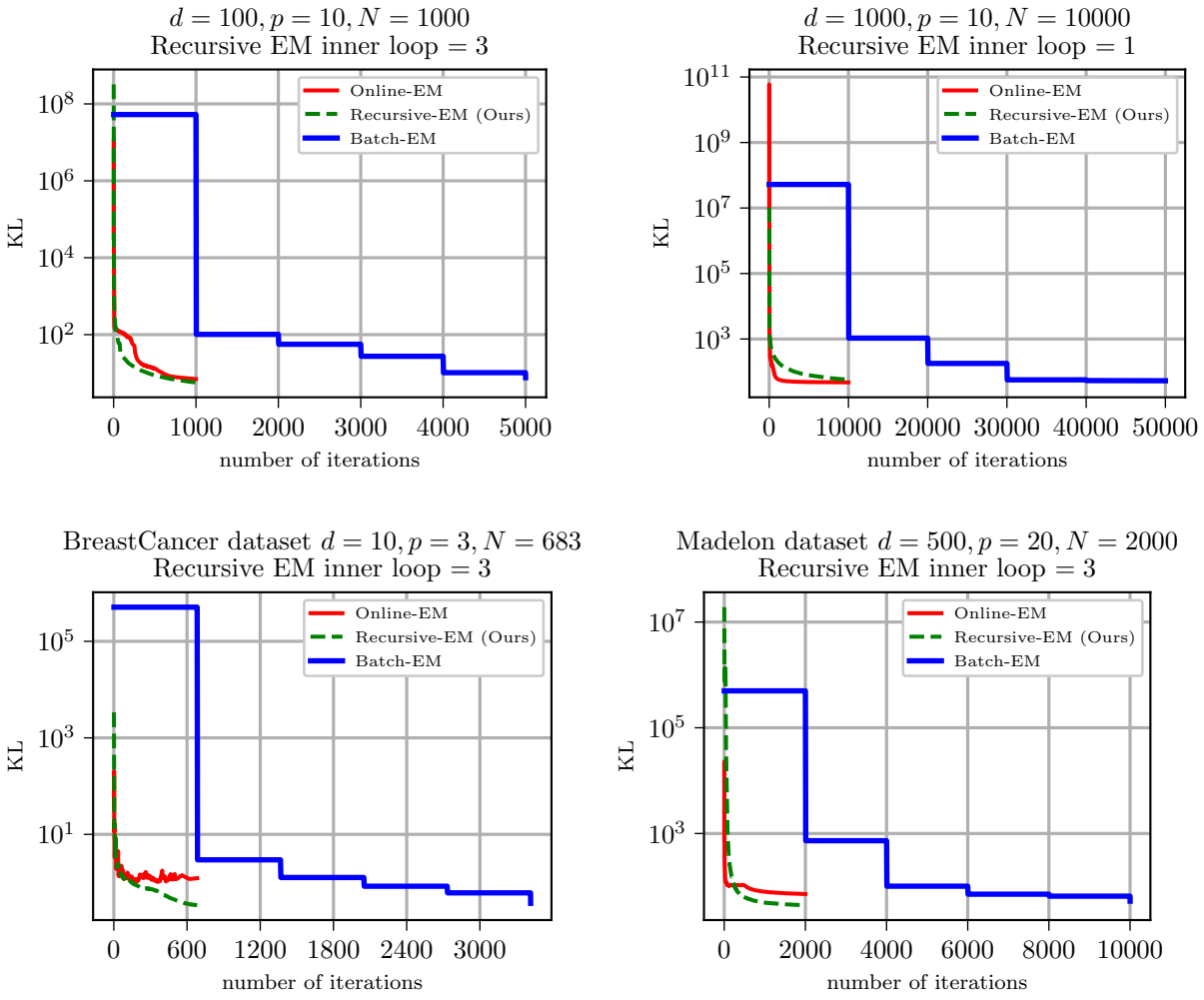


Figure 1: Factorization of a synthetic large scale covariance matrix of dimension $d = 100$ (upper left plot) and $d = 1000$ (upper right plot). The lower plots show additional results on two covariance matrices computed from the Madelon and Breast Cancer datasets. We show in dash green line our recursive EM algorithm, which is parameter free, and in red plain line the online EM tuned with a learning rate $\gamma = \frac{1}{t^{0.6}}$ (see Appendix C for implementation details). We show also the batch EM, in solid blue line, which uses 5 passes through the entire dataset, contrary to the online variants which are single pass. With 3 inner loops in dimension $d = 100$ (upper left plot), the recursive EM is clearly superior to the online EM and to the batch EM after the first pass. In higher dimension (upper right plot), we achieve a similar loss to the online EM using only 1 inner loop in the recursive EM. Moreover, the recursive EM gives better results than the other algorithms on real datasets as illustrated in lower left plot and lower right plot. Finally, note that the comparison between the online and batch methods should be understood in the sense that batch estimation requires processing all the data several times, (albeit more efficient, since 5 steps are enough to converge).

The linear Kalman filter is a parameter-free online algorithm which gives the exact solution of a linear regression problem after only one pass on the dataset, that is, it recursively computes exactly the posterior given all the data it has considered so far. However, it requires to estimate online the

| Algorithm | Nb. iterations | Time per iteration | Memory cost |
|---|---|---|---|
| Batch EM (estimated) | Fixed ($\ll N$) | - | **8000 GB** |
| Online EM-1 inner loop | $N$ | 1 s | **1.6 GB** |
| Recursive EM-1 inner loop | $N$ | 2 s | **0.8 GB** |

Table 1: **Memory test for large scale matrix factorization with $d = 10^6$, and $p = 100$**: Test dedicated for memory requirement, the recursive and online EM have been executed on a laptop (Intel core i7 at 2.3 GHz on CPU) in dimension one million, the memory cost for the batch EM have been estimated as it scale quadratically in $d$ and did not fit the memory, so no time is available for it. A reduced number of samples $N$ have been considered on this experiment since they do not influence the memory cost. Recursive EM require much operations by iteration than online EM but consume less memory than online EM and much less than batch EM.

$d \times d$ covariance matrix of the inputs, which is intractable in high dimension. In this experiment, this matrix is approximated with the recursive EM algorithm to provide large scale linear regression.

The inputs $x_t$ are generated using the following Gaussian distribution:

$$x_t \sim \mathcal{N}(0, C) \quad \text{with} \quad C = M^T \text{Diag}(1, 1/2^c, \cdots, 1/d^c) M, \tag{49}$$

where $M$ is an orthogonal rotation matrix and $c$ a coefficient driving the condition number ($c = 1$ by default). Since the matrix $C$ is rotated, it is not directly available in a factor analysis form. The inputs are normalized in mean.

The outputs $y_t$ are generated with a normal noise:

$$y_t = x_t^T \theta^* + \varepsilon \text{ with } \varepsilon \sim \mathcal{N}(0, 1), \tag{50}$$

We consider the Bayesian setting where $\theta$ is supposed to be a Gaussian random value and we want to estimate the parameters of its distribution $\mathcal{N}(\mu, P)$ where $P^{-1} = WW^T + \Psi$, given all the examples $(x_t, y_t)$ seen so far. We suppose that the prior follows a centered isotropic Gaussian distribution $\theta_0 \sim \mathcal{N}(0, \sigma_0^2 \mathbb{I}_d)$ and compute the initial values of the factor analysis parameters $W_0$ and $\Psi_0$ as detailed in Section 5.1.2. Our recursive scheme is sensitive to the parameter $\sigma_0$: a high value of $\sigma_0$ (flat prior) may lead to bad conditioning at first steps and make our recursive algorithm diverge. A low value for $\sigma_0$ (strong prior) speeds up the convergence of the algorithm. This value defines the shape of the posterior and the difficulty of the estimation problem and can be used to test the robustness of our algorithm in the following experiments. In this Section we consider a prior deviation $\sigma_0 = 1$.

When $d = p$ we have checked that our algorithm matches exactly the linear Kalman filter's estimates. To assess how the factor analysis approximation degrades the results, we have made experiments for different values of the latent space dimension $p$. We see in Figure 2 that the divergence decreases globally for all latent dimensions even if for lower values of $p$ the divergence may temporary increase. As expected the convergence is faster for higher values of $p$. Even when reducing the dimension by a factor 200, we still obtain a solution quite close to the optimal one given by the full linear Kalman filter. The memory cost for higher dimension is detailed in Table 2.

### 5.3 Application to Logistic regression

We now apply the method to derive a limited-memory version on a logistic regression problem which is a particular case of the generalized linear model described in Section 4.1. In this model, we seek to learn the parameter $\theta$ encoding the hyperplane from $N$ examples $(x_t, y_t)$. We consider the Bayesian

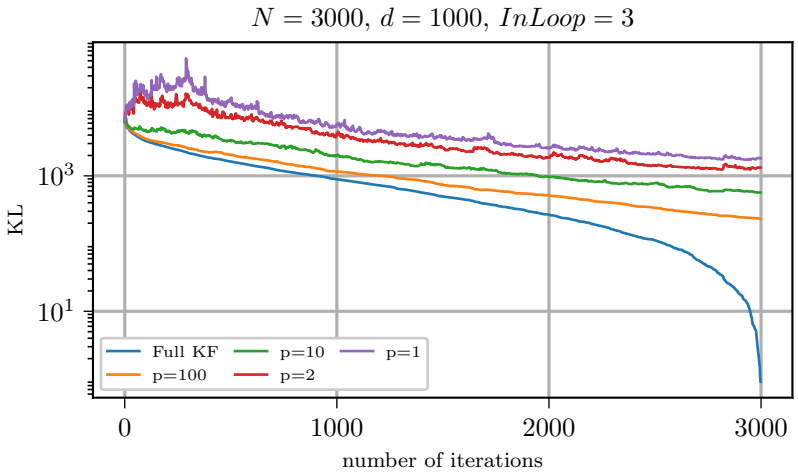


Figure 2: Sensitivity to the latent dimension $p$. The full Kalman error converges exactly to 0, as the Kalman filter exactly finds the posterior. The approximated version with $p = 100$ runs a close second. When $p$ is very low, oscillations occur.

| Algorithm | Nb. iterations | Time per iteration (for 3 inner loops) | Memory cost |
|---|---|---|---|
| Full Kalman | $N$ | - | **8000 GB** |
| L-RVGA ($p = 100$) | $N$ | 6 s | **816 MB** |
| L-RVGA ($p = 10$) | $N$ | 0.7 s | **96 MB** |
| L-RVGA ($p = 2$) | $N$ | 0.2 s | **32 MB** |
| L-RVGA ($p = 1$) | $N$ | 0.06 s | **24 MB** |

Table 2: **Memory requirements test for linear regression with $d = 10^6$ and variable $p$**: The L-RVGA has been executed on a laptop (Intel core i7 at 2.3 GHz on CPU) in dimension one million, the memory cost for the full Kalman has been estimated as it scales quadratically in $d$ and did not fit into the memory, so that execution time is not available for it. A reduced number of samples $N$ has been considered on this experiment since they do not influence the memory cost. When $p$ is low the cost in memory as well as the time per iteration (with 3 inner loops for the recursive EM) is drastically reduced.

setting where $\theta \sim \mathcal{N}(\mu, P)$ and the observations are generated from $y_t = \sigma(\mu^T x_t)$, where $\sigma$ denotes here the logistic function. The estimation of $\mu_t$ and $P_t$ are given by the RVGA updates for general linear model (29)-(30). These updates involve the expectation of the gradient and the Hessian of the logistic loss which can be approximated as follow:

$$\mathbb{E}_\theta[\nabla_\theta \ell_t(\theta)] = -y_t x_t + \mathbb{E}_\theta[\sigma(x_t^T\theta)]x_t \approx -y_t x_t + \sigma(k_t x_t^T \mu_t) \tag{51}$$

$$\mathbb{E}_\theta[\nabla^2_\theta \ell_t(\theta)] = \mathbb{E}_\theta[\sigma(x_t^T\theta)(1-\sigma(x_t^T\theta))]x_t x_t^T \approx k_t \sigma'(k_t x_t^T \mu_t) x_t x_t^T \tag{52}$$

$$\text{where } k_t = \frac{\beta}{\sqrt{x_t^T P_t x_t + \beta^2}} \text{ and } \beta = \sqrt{\frac{8}{\pi}}. \tag{53}$$

These equations were derived in our previous work (see (Lambert et al., 2021), Section 4.1) and come from the approximation of the logistic function $\sigma$ with the inverse probit function $\phi$ (Barber

and Bishop, 1998a) :$\sigma(x) \approx \phi(\frac{1}{\beta}x) = \frac{1}{2}(1 + erf(\frac{x}{\sqrt{2}\beta}))$. We can then rewrite (29)-(30) as:

$$\mu_t = \mu_{t-1} + P_{t-1}x_t(y_t - \sigma(k_t x_t^T \mu_t)) \tag{54}$$

$$P_t^{-1} = P_{t-1}^{-1} + k_t \sigma'(k_t x_t^T \mu_t) x_t x_t^T. \tag{55}$$

The scheme is now implicit owing only to the two scalar parameters $\nu = x_t^T P_t x_t$ and $\alpha = x_t^T \mu_t$ which can efficiently be computed by solving a scalar fixed point equation with a Newton solver, see Lambert et al. 2021, Section 4.2.

We can in turn apply our recursive EM algorithm on the input $X_t := x_t \sqrt{k_t \sigma'(k_t x_t^T \mu_t)}$ for high-dimensional logistic regression. To assess the performance, we generate $N$ synthetic pairs $(x_t, y_t)$ where the inputs are generated from the same Gaussian distribution than in the linear regression case. The inputs are normalized in mean and the prior is set to $\sigma_0 = 4$ which corresponds to a sharp posterior distribution. In Figure 3 we plot the KL divergence between our current Gaussian estimate $\mathcal{N}(\mu_t, P_t)$ and the true posterior, that is $KL\left(\mathcal{N}(\mu_t, P_t)||p(\theta|y_1, x_1, \dots y_N, x_N)\right)$. In logistic regression, we have a simple expression for the posterior at each $\theta$, up to a normalizing constant. As the former left KL divergence is an expectation under $\mathcal{N}(\mu_t, P_t)$, it may be approximated via Monte-Carlo sampling. This offers a way to perform relative comparisons between the algorithms in terms divergence with respect to the true (unnormalized) posterior. For more details see Lambert et al. 2021.

We compare the results for different values of the parameter $p$ with the Laplace approximation which gives a batch approximation of $\mu$ and $P$. The L-RVGA converges to the batch Laplace approximation and may even yield lower divergence. This is because the covariance given by the Laplace approximation spills out the true posterior to regions of very low probability whereas the L-RVGA avoids them. Contrary to the linear case, even with $p = 1$, we obtain very good results. The memory requirements in dimension one million yields identical costs to those reported in Table 2 for linear regression.

### 5.4 General nonlinear method evaluation

We address in this section the general nonlinear L-RVGA, based on the approximations detailed in Section 4.2. To assess the method, we choose to apply the general nonlinear method to logistic regression, again, as we have closed-form expressions that may serve as ground truth.

As the generalized Gauss-Newton approximation, presented in Section (4.2.3), is always exact in the logistic case, only the other approximations will be evaluated: sampling to approximate the expectation, extra-gradient to approximate the implicit scheme, and factor analysis to approximate the covariance matrix.

As logistic regression is based on a generalized linear model, $\mu_t$ and $P_t$ are given by the RVGA updates for generalized linear models (29)-(30). These updates involve the expectation of the gradient and the Hessian of the logistic loss:

$$\mathbb{E}_\theta[\nabla_\theta \ell_t(\theta)] = -y_t x_t + \mathbb{E}_\theta[\sigma(x_t^T \theta)] x_t \tag{56}$$

$$\mathbb{E}_\theta[\nabla_\theta^2 \ell_t(\theta)] = \mathbb{E}_\theta[\sigma(x_t^T \theta)(1 - \sigma(x_t^T \theta))] x_t x_t^T. \tag{57}$$

Previously, we have seen those expectations could be easily computed, see (51)-(52) and following text. For comparison purposes, we may on the other hand approximate those expectations using

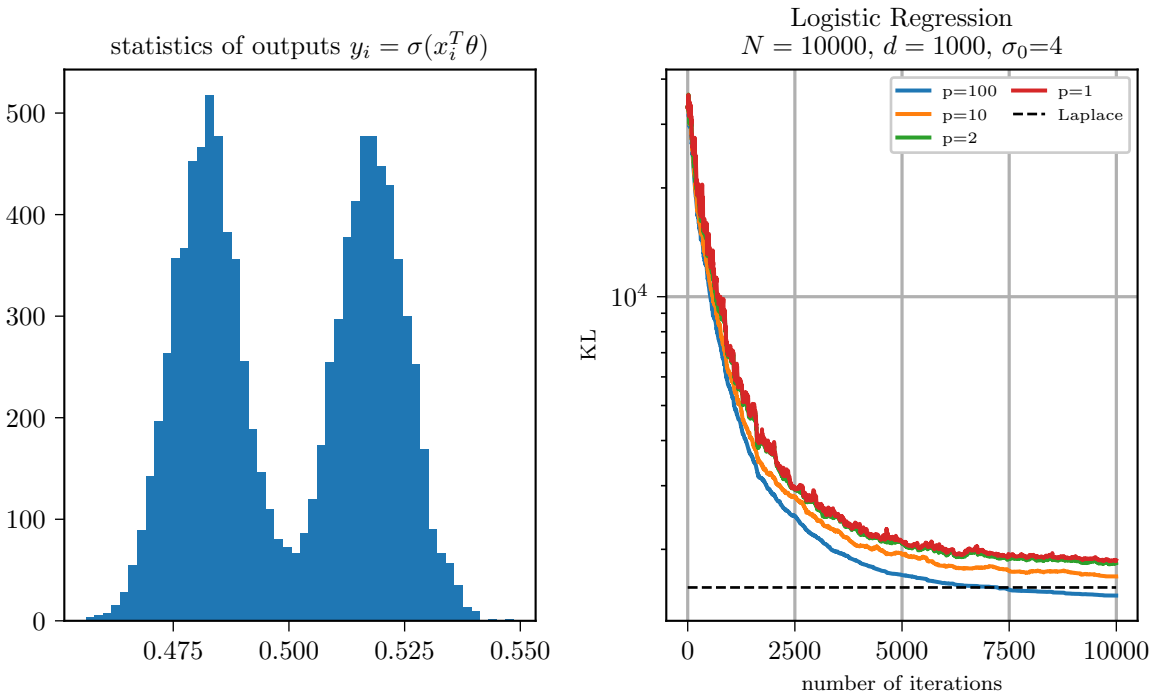


Figure 3: Sensitivity to the latent dimension $p$. The L-RVGA algorithms may provide a a lower divergence than the batch Laplace approximation in the sense of KL divergence to the true posterior. The estimated mean is roughly aligned with the maximum a posteriori (MAP). Only one inner loop proves sufficient to ensure convergence as concerns logistic regression. The prior is set to be $P_0 = \sigma_0 \mathbb{I}_d$ with $\sigma_0 = 4$ and the algorithm uses it for initialization. The KL divergence is computed with sampling and is unnormalized.

the general sampling procedure of Section 4.2.2, that is,

$$\mathbb{E}_\theta[\sigma(x_t^T\theta)] \approx \frac{1}{K}\sum_{i=1}^{K} \sigma(x_t^T\theta_i) \tag{58}$$

$$\mathbb{E}_\theta[\sigma(x_t^T\theta)(1-\sigma(x_t^T\theta))] \approx \frac{1}{K}\sum_{i=1}^{K} \sigma(x_t^T\theta_i)(1-\sigma(x_t^T\theta_i)),$$

where the samples $\theta_i$ are drawn from the Gaussian distribution $\mathcal{N}(\mu, (\Psi + WW^T)^{-1})$ exactly using ensemble sampling as described in Section 4.2.2. Table 3 shows that the method approximates well the closed from expression used in Section 5.3.

We then combined sampling with the extra-gradient proposed method, where we skip the extra covariance update, as described in Section 4.2.1, since we observed it helped convergence. We found few samples may be sufficient to provide good convergence of the algorithm in terms of KL divergence, as shown in Figure 4.

**Sources:** The sources of the code are available on Github on the following repository: `https://github.com/marc-h-lambert/L-RVGA`.

| Method | $\mathrm{Tr}(WW^T + \Psi)^{-1}$ | $\mathbb{E}_\theta[\sigma(x_t^T\theta)]$ |
|---|---|---|
| Baseline | 9.24 | 0.773**801** |
| Cholesky sampling | 9.66 | 0.773**857** |
| **Ensemble sampling** | 9.53 | 0.773**855** |

Table 3: **Test for the ensemble sampling approximation in dimension $d = 10000$ and $p = 10$**: Approximation of the expectation $\mathbb{E}_\theta[\sigma(x_t^T\theta)]$ is performed with 10 samples. The closed-form expression with inverse probit approximation is considered as the baseline. We show also how the trace of the matrix $(WW^T + \Psi)^{-1}$ is approximated. The Cholesky sampling uses the actual square root decomposition of $(WW^T + \Psi)^{-1}$. The ensemble sampling is our method described in Section 4.2.2. It achieves similar results as Cholesky sampling which is not memory-limited since it uses the full matrix and inverts it.

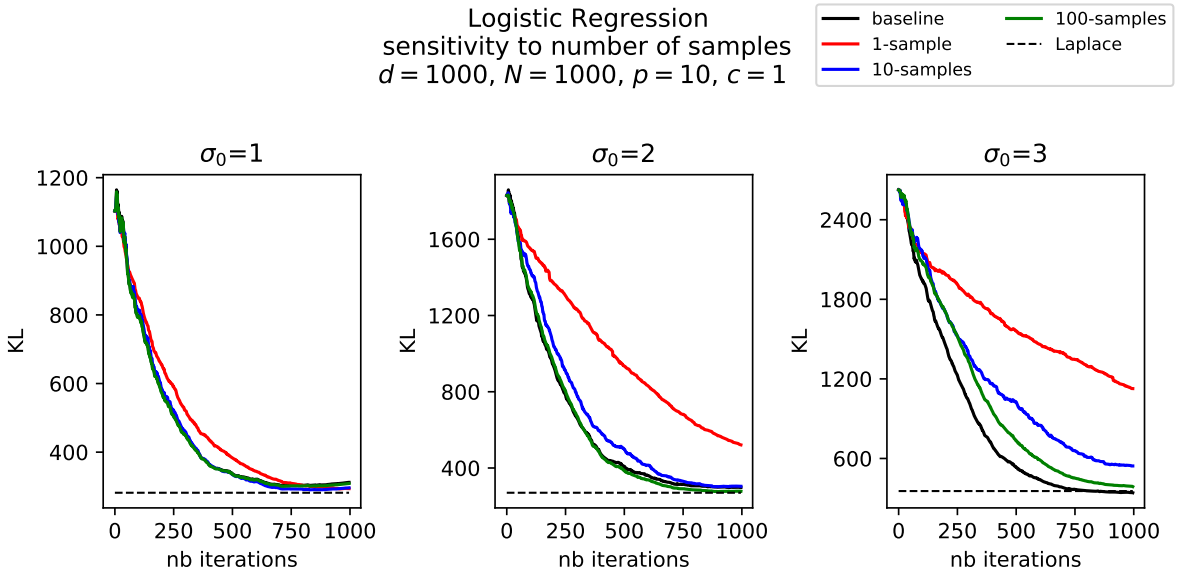


Figure 4: Approximation of the logistic posterior for different deviations on the priors $\sigma_0 = 1, 2$ and 3. The inputs have been normalized in mean. We compare the L-RVGA where the expectations are computed analytically (baseline) and the L-RVGA where the expectations are approximated with sampling for 1, 10 and 100 samples. All the algorithms use a Mirror prox (where we have skipped the extra covariance update) and use the same factor analysis approximation $p = 10$. We see K = 10 samples are sufficient even in the more difficult case to converge to the batch Laplace KL.

## Conclusion

We have developed a new second order algorithm, called L-RVGA, for online variational inference which scales to both large data sets and high dimension. This algorithm is based on a two-stage variational problem which combines a variational Gaussian approximation followed by a factor analysis approximation of the inverse of the covariance matrix. L-RVGA is able to estimate the mean and the covariance of the distribution of the latent parameters in a memory-efficient and parameter-free way and with only one pass through the data. We have tested it on linear and logistic regression problems and shown how to extend it to more general nonlinear problems with extra-gradients, memory-efficient sampling and the generalized Gauss-Newton approximation. To build our generic algorithm we have introduced two new tools: a recursive EM algorithm for factor

analysis which is parameter-free and faster than the online EM in this context; and a new sampler for Gaussian distribution with a structured precision matrix.

Beyond variational inference, we anticipate the L-RVGA may prove useful for stochastic optimization keeping only the mean and using the covariance to build an adaptive learning step. This version could be compared to state-of-the-art algorithms in limited memory optimization such as Adagrad or even L-BFGS. We will investigate this direction in future work and will extend our algorithm using adaptive filtering techniques to further increase its performance. We believe that a deeper connection between the communities of stochastic optimization and Kalman filtering may bring new ideas to tackle nonlinear stochastic problems.

### Acknowledgements

This work was funded by the French Defence procurement agency (DGA) and by the French government under management of Agence Nationale de la Recherche as part of the "Investissements d'avenir" program, reference ANR-19-P3IA-0001(PRAIRIE 3IA Institute). We also acknowledge support from the European Research Council (grant SEQUOIA 724063).

## References

Sivaram Ambikasaran, Michael O'Neil, and Karan Raj Singh. Fast symmetric factorization of hierarchical matrices with applications. 2014. doi: 10.48550/ARXIV.1405.0223.

Christophe Andrieu, Nando de Freitas, Arnaud Doucet, and Michael I. Jordan. An introduction to MCMC for machine learning. *Machine Learning*, 50:5–43, 2003.

Francis Bach and Kfir Y. Levy. A universal algorithm for variational inequalities adaptive to smoothness and noise. *Conference on learning theory*, 2019.

D. Barber. *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2011.

D. Barber and Christopher Bishop. Ensemble learning in bayesian neural networks. In *Generalization in Neural Networks and Machine Learning*, pages 215–237, 1998a.

David Barber and Christopher Bishop. Ensemble learning for multi-layer networks. *Advances in Neural Information Processing Systems*, 10, 1998b.

Olivier Cappé and Eric Moulines. Online expectation-maximization algorithm for latent data models. *Journal of the Royal Statistical Society*, 71(3):593–613, 2009.

Edward Challis and David Barber. Gaussian Kullback-Leibler approximate inference. *Journal of Machine Learning Research*, 14:2239–2286, 2013.

Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.

I. Csiszár and P.C. Shields. Information theory and statistics: A tutorial. *Foundations and Trends in Communications and Information Theory*, 1(4):417–528, 2004.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society*, 39:1–38, 1977.

Rubin Donald and Thayer Dorothy. *EM algorithms for ML factor analysis*, volume vol. 47(1),. The Psychometric Society, 1982.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

Geir Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics. *J. Geophys. Res*, 99:10143–10162, 1994.

Geoffrey E. Hinton and Drew van Camp. Keeping the neural networks simple by minimizing the description length of the weights. *Annual Conference on Computational Learning Theory*, pages 11–18, 1993.

A.H. Jazwinski. *Stochastic Processes and Filtering Theory*. Elsevier Science, 1970.

Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. *An Introduction to Variational Methods for Graphical Models*, pages 105–161. Springer Netherlands, 1998.

Mohammad Emtiyaz Khan, Didrik Nielsen, Voot Tangkaratt, Wu Lin, Yarin Gal, and Akash Srivastava. Fast and scalable bayesian deep learning by weight-perturbation in Adam. *arXiv:1806.04854*, 2018.

Frederik Kunstner, Philipp Hennig, and Lukas Balles. Limitations of the empirical Fisher approximation for Natural gradient descent. *Advances in Neural Information Processing Systems*, pages 4156–4167, 2019.

Marc Lambert, Silvère Bonnabel, and Francis Bach. The recursive variational gaussian approximation (r-vga). *Statistics and Computing*, 32(1):10, 2021.

Wu Lin, Mohammad Emtiyaz Khan, and Mark Schmidt. Stein's lemma for the reparameterization trick with exponential family mixtures. *arXiv preprint arXiv:1910.13398*, 2019.

S. L Tan Linda and J. Nott David. Variational inference for generalized linear mixed models using partially noncentered parametrizations. *arxiv:1205.3906v3*, 2013.

James Martens. New insights and perspectives on the natural gradient method. *arXiv preprint arXiv:1412.1193*, 2014.

Tran Minh-Ngoc, J. Nott David, and Kohn Robert. Variational bayes with intractable likelihood. *arxiv:1503.08621v2*, 2016.

Aaron Mishkin, Frederik Kunstner, Didrik Nielsen, Mark W. Schmidt, and Mohammad Emtiyaz Khan. Slang: Fast structured covariance approximations for bayesian deep learning with natural gradient. *Advances in Neural Information Processing Systems*, pages 6248–6258, 2018.

R. Neal and G. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. *Learning in graphical models*, pages 355–368, 1999.

Arkadi Nemirovski. Prox-method with rate of convergence $O(1/t)$ for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM J. on Optimization*, 15(1):229–251, 2005.

Yann Ollivier. Online Natural gradient as a Kalman filter. *Electronic Journal of Statistics*, 12: 2930–2961, 2018.

Victor M.H. Ong, David J. Nott, and Michael S Smith. Gaussian variational approximation with a factor covariance structure. *Journal of Computational and Graphical Statistics*, 27:465–478, 2018.

Manfred Opper and Cédric Archambeau. The variational Gaussian approximation revisited. *Neural computation*, 21:786–792, 2009.

Dinh Tuan Pham, Jacques Verron, and Marie Christine Roubaud. A singular evolutive extended Kalman filter for data assimilation in oceanography. *Journal of Marine Systems*, 16(3):323 – 340, 1998.

T. S. Pitcher. Review: O. Barndorff-Nielsen, Information and exponential families in statistical theory. *Bulletin (New Series) of the American Mathematical Society*, 1(4):667 – 668, 1979.

Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. *Artificial intelligence and statistics*, pages 814–822, 2014.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

Nicolas L. Roux, Pierre antoine Manzagol, and Yoshua Bengio. Topmoumoute online natural gradient algorithm. *Advances in Neural Information Processing Systems*, 20:849–856, 2008.

Sam Roweis. EM algorithms for PCA and SPCA. *Advances in Neural Information Processing Systems*, 10, 1998.

Michael E. Tipping and Christopher M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society.*, 61:611–622, 1999.

# A The recursive EM

## A.1 Derivation of the fixed point equation for factor analysis with EM

In this Section we show that finding the factor analysis parameter with EM is equivalent to iterate through the fixed point equation (20). In the context of probabilistic principal component analysis Tipping and Bishop (1999) have highlighted (in their Appendix $B$) how the EM can be rewritten as a fixed point equation. The following proof is an extension of this result to the factor analysis case.

In factor analysis, we want to approximate the empirical covariance matrix $\frac{1}{K}\sum_{i=1}^{K} v_i v_i^T$ with a "diagonal + low rank" structure $\Psi + WW^T$. In the classical EM approach we introduce latent variables $z_i$ such that our samples can be rewritten $v_i = Wz_i + \varepsilon_i$ where $\varepsilon_i \sim \mathcal{N}(0, \Psi)$. We note $V = \begin{pmatrix} v_1 \cdots v_K \end{pmatrix}$ the sample matrix of size $d \times K$ such that $\frac{1}{K}\sum_{i=1}^{K} v_i v_i^T = \frac{1}{K} VV^T$ and $Z = \begin{pmatrix} z_1 \cdots z_K \end{pmatrix}$ the latent matrix of size $p \times K$.

At the expectation step (E-step), the latent variables $z_i$ are estimated conditionally on the observation and the current parameter estimates using the conditioning formula:

$$p(z_i|v_i, W, \Psi) = \mathcal{N}(M^{-1}W^T\Psi^{-1}v_i, M^{-1}) \text{ where } M = \mathbb{I}_p + W^T\Psi^{-1}W. \tag{59}$$

At the maximization step (M-step), the latent variables are assumed fixed and the parameters are adjusted to maximize the total likelihood defined by:

$$L(W,\Psi) = \log p(V,Z|W,\Psi) = \sum_{i=1}^{K} \log p(v_i|z_i,W,\Psi) + \sum_{i=1}^{K} \log p(z_i) \tag{60}$$

$$= -\frac{K}{2}(v_i - Wz_i)^T\Psi^{-1}(v_i - Wz_i) - \frac{K}{2}\log\det\Psi - \frac{K}{2}\log(2\pi) + \sum_{i=1}^{K}\log p(z_i). \tag{61}$$

The derivative of the expectation of $L$ with respect to $W$ gives:

$$\frac{\partial}{\partial W}\mathbb{E}_{Z\sim p(Z|V)}[L(W,\Psi)] = -\sum_{i=1}^{K}\Psi^{-1}W\mathbb{E}[z_i z_i^T|v_i] - \sum_{i=1}^{K}\Psi^{-1}v_i\mathbb{E}[z_i|v_i]^T = 0, \tag{62}$$

and we obtain the optimum $W^{(n)} = \sum_{i=1}^{K} v_i\mathbb{E}[z_i|v_i]^T(\sum_{i=1}^{K}\mathbb{E}[z_t z_t^T|v_i])^{-1}$, where the $^{(n)}$ stands for "new".

Taking the derivative of the expectation of $L$ with respect to $\Psi^{-1}$ and keeping only the diagonal terms yields:

$$\Psi^{(n)} = \mathrm{diag}(\frac{1}{K}\sum_{i=1}^{K} v_i v_i^T + \frac{1}{K}\sum_{i=1}^{K} W^{(n)}\mathbb{E}[z_i z_i^T|v_i]W^{(n)^T} - \frac{2}{K}\sum_{i=1}^{K} W^{(n)}\mathbb{E}[z_i|v_i]v_i^T) \tag{63}$$

$$= \mathrm{diag}(\frac{1}{K}\sum_{i=1}^{K} v_i v_i^T - W^{(n)}\frac{1}{K}\sum_{i=1}^{K}\mathbb{E}[z_i|v_i]v_i^T), \tag{64}$$

where we have replaced $W^{(n)^T}$ by its optimal value in equation (63).

And finally the EM updates gives:

(65)

**E-Step:**

$$\mathbb{E}[z_i|v_i] = M^{-1}W^T\Psi^{-1}v_i$$
$$\mathbb{E}[z_i z_i^T|v_i] = M^{-1} + \mathbb{E}[z_i|v_i]\mathbb{E}[z_i|v_i]^T. \tag{66}$$

**M-Step:**

$$W^{(n)} = \sum_{i=1}^{K} v_i\mathbb{E}[z_i|v_i]^T(\sum_{i=1}^{K}\mathbb{E}[z_i z_i^T|v_i])^{-1}$$
$$\Psi^{(n)} = \mathrm{diag}(\frac{1}{K}\sum_{i=1}^{K} v_i v_i^T - W^{(n)}\frac{1}{K}\sum_{i=1}^{K}\mathbb{E}[z_i|v_i]v_i^T). \tag{67}$$

The EM updates (66) and (67) can be rewritten in batch form using the matrix notation $V$ and $Z$:

(68)

**E-Step:**

$$\mathbb{E}[Z|V] = M^{-1}W^T\Psi^{-1}V$$
$$\sum_{i=1}^{K}\mathbb{E}[z_i z_i^T|v_i] := \mathbb{E}[ZZ^T|V] = KM^{-1} + \mathbb{E}[Z|V]\mathbb{E}[Z|V]^T = K(M^{-1} + M^{-1}W^T\Psi^{-1}S_K\Psi^{-1}WM^{-1}). \tag{69}$$

**M-Step:**

$$W^{(n)} = V\mathbb{E}[Z|V]^T\mathbb{E}[ZZ^T|V]^{-1}$$
$$\Psi^{(n)} = \mathrm{diag}(\frac{1}{K}VV^T - W^{(n)}\frac{1}{K}\mathbb{E}[Z|V]V^T). \tag{70}$$

And finally the E-step and M-step can be fused to form a fixed-point equation:

$$\begin{aligned}
W^{(n)} &= V\mathbb{E}[Z|V]^T\mathbb{E}[ZZ^T|V]^{-1} \\
&= S_K\Psi^{-1}WM^{-1}(M^{-1} + M^{-1}W^T\Psi^{-1}S_K\Psi^{-1}WM^{-1})^{-1} \\
&= S_K\Psi^{-1}W(\mathbb{I}_p + M^{-1}W^T\Psi^{-1}S_K\Psi^{-1}W)^{-1} \\
\Psi^{(n)} &= \mathrm{diag}(\frac{1}{\mathrm{K}}\mathrm{V}\mathrm{V}^{\mathrm{T}} - \mathrm{W}^{(\mathrm{n})}\frac{1}{\mathrm{K}}\mathbb{E}[\mathrm{Z}|\mathrm{V}]\mathrm{V}^{\mathrm{T}}) \\
&= \mathrm{diag}(\mathrm{S_K} - \mathrm{W}^{(\mathrm{n})}\mathrm{M}^{-1}\mathrm{W}^{\mathrm{T}}\Psi^{-1}\mathrm{S_K}).
\end{aligned}$$

which is the fixed point equation given in (20).

# B The fixed point EM is equivalent to the MLE fixed point

In this section we show new results concerning the equivalence of the fixed points for the marginal likelihood (MLE algorithm) and the total likelihood (EM algorithm) for the particular case of factor analysis. We consider here the batch factor analysis problem. This result may not be applied to our RVGA algorithm but for each inner loop of it which are guaranteed to increase the one sample likelihood.

The marginal likelihood and the total likelihood are both related as follows:

$$\max_{W,\Psi} \quad \log p(v_1,\ldots,v_N|W,\Psi) \text{ where } S_N = \frac{1}{N}\sum_{i=1}^{N} v_i v_i^T = P^{-1} \quad \text{(MLE)} . \tag{71}$$

$$\leq \max_{W,\Psi} \quad \mathbb{E}_z[\log p(v_1,\ldots,v_N,z|W,\Psi)] \text{ with } p(v \mid z) \sim \mathcal{N}(Wz,\Psi),\ p(z) \sim \mathcal{N}(0,I_p) \quad \text{(EM)} . \tag{72}$$

The advantage of the EM approach is that the total likelihood appearing in (72) is guaranteed to increase at each step, i.e., the algorithm is stable. The EM algorithm may not always converge to the maximum of the marginal likelihood appearing in (71) but converges to a stationary point which turns out to be also a stationary point for the maximum likelihood. This fact was already highlighted by Neal and Hinton (1999) and Cappé and Moulines (2009) but we specify the result in the case of factor analysis in an algebraic way. We first write the maximum likelihood as a fixed point, then derivate an equivalence between the two fixed points. This result is finally used to derivate an equivalence between different eigenvalues decomposition algorithms.

## B.1 Factor analysis with maximum likelihood (MLE) as a fixed point.

In this section we writes the maximum likelihood on the factor analysis parameters as a fixed point equation. This is a well known result derived from the chapter 21.2 of the book of Barber (2011) but we want to use this result to make a connexion to the fixed point obtained with EM. We want to approximate a matrix $S$ with a matrix $WW^T + \Psi$ by maximizing the following likelihood (where $C$ is a constant):

$$\max_{W,\Psi} \quad L(W,\Psi) = \max_{W,\Psi} \quad -\frac{N}{2}\mathrm{Tr}((WW^T+\Psi)^{-1}S_N) - \frac{N}{2}\log\det(WW^T+\Psi) + C. \tag{73}$$

A necessary condition on the optimal solution is to annulate the gradients:

$$\frac{\partial L(W,\Psi)}{\partial W} = (WW^T+\Psi)^{-1}W^T - (WW^T+\Psi)^{-1}S_N(WW^T+\Psi)^{-1}W = 0 \tag{74}$$

$$\frac{\partial L(W,\Psi)}{\partial \Psi} = (WW^T+\Psi)^{-1} - (WW^T+\Psi)^{-1}S_N(WW^T+\Psi)^{-1} = 0, \tag{75}$$

leading to the fixed-point equations:

**Fixed-point equations for the MLE** (76)

$$W^{(n)} = S_N(WW^T+\Psi)^{-1}W \tag{77}$$

$$\Psi = \mathrm{diag}(\mathrm{S_N} - \mathrm{W^{(n)}W^{(n)^T}}). \tag{78}$$

## B.2 Equivalence of fixed points and convergence

The following proposition shows the algebraic equivalence of the fixed points.

**Proposition 1.** *The factor analysis parameters which maximize the marginal likelihood, that is, the solution to* (71) *satisfy the following fixed point equation:*

$$W^{(n)} = S_N(WW^T+\Psi)^{-1}W \tag{79}$$

$$\Psi = \mathrm{diag}(S_N - W^{(n)}{W^{(n)}}^T). \tag{80}$$

*This fixed point equation is equivalent to the EM fixed-point equation:*

$$W^{(n)} = S_N\Psi^{-1}W(\mathbb{I}_p + M^{-1}W^T\Psi^{-1}S_N\Psi^{-1}W)^{-1} \tag{81}$$

$$\textit{where: } M = \mathbb{I}_p + W^T\Psi^{-1}W \tag{82}$$

$$\Psi = \mathrm{diag}(S_N - W^{(n)}M^{-1}W^T\Psi^{-1}S_N). \tag{83}$$

*As a consequence, the EM algorithm converges to a stationary point which is also a stationary point for the likelihood.*

*Proof.* The proof for the equivalence of the fixed points is straightforward, the update for $W$ can be rewritten as:

$$W = S_N(WW^T+\Psi)^{-1}W \tag{84}$$

$$= S_N\Psi^{-1}W(\mathbb{I}_p + W^T\Psi^{-1}W)^{-1} \text{ (From the Woodbury formula)} \tag{85}$$

$$= S_N\Psi^{-1}WM^{-1} \tag{86}$$

$$= S_N\Psi^{-1}W(\mathbb{I}_p + (S_N\Psi^{-1}WM^{-1})^T\Psi^{-1}W)^{-1} \tag{87}$$

$$= S_N\Psi^{-1}W(\mathbb{I}_p + M^{-1}W^T\Psi^{-1}S_N\Psi^{-1}W)^{-1}, \tag{88}$$

where we have replaced the term $W^T$ in (85) by its development in (86).

The update for $\Psi$ can be rewritten as:

$$\Psi = \mathrm{diag}(\mathrm{S_N} - \mathrm{WW^T}) \tag{89}$$
$$= \mathrm{diag}(S_N - W(S_N\Psi^{-1}WM^{-1})^T) \tag{90}$$

where we have replaced the term $W^T$ in (89) by its development in (86), leading to

$$\Psi = \mathrm{diag}(S_N - WM^{-1}W^T\Psi^{-1}S_N) \tag{91}$$
$$= \mathrm{diag}(S_N - W_nM^{-1}W^T\Psi^{-1}S_N). \tag{92}$$

□

### B.3 Relation to singular value decomposition

The fixed point equation from the maximum likelihood is solved using a singular value decomposition (SVD) of $S_N$ (see Barber (2011), chapter 21.2). The equivalence with the EM fixed point suggest that the EM make implicitly an SVD decomposition. It can be shown it is the case if we consider an asymptotical Probabilistic Principal Component analysis form (PPCA), ie $\Psi = \sigma I$ where we let tends the parameter $\sigma$ to 0. This results was shown by Roweis (1998) for the fixed point EM with PPCA and is related to the MLE fixed point in the following Corollary.

**Corollary 1** (of Prop. 1)**.**

*The factor analysis MLE fixed-point equation for PPCA:*

$$W^{(n)} = S_N(WW^T + \sigma^2\mathbb{I}_d)^{-1}W = S_NW(\sigma^2\mathbb{I}_p + W^TW)^{-1} \tag{93}$$

*converge for $\sigma \to 0$ to a fixed point :*

$$W^{(n)} = S_NW(W^TW)^{-1}, \tag{94}$$

*which correspond to the power method:*

$$w \leftarrow S_N\frac{w}{||w||^2} \quad \textit{(given here in vectorial form).} \tag{95}$$

*The factor analysis EM fixed-point for PPCA:*

$$W_n = S_NW(\sigma^2\mathbb{I}_p + (\sigma^2\mathbb{I}_p + W^TW)^{-1}W^TS_NW)^{-1} \tag{96}$$

*converge for $\sigma \to 0$ to a fixed point :*

$$W^{(n)} = S_NW(W^TS_NW)^{-1}W^TW, \tag{97}$$

*which is equivalent to the EM-PCA method (Roweis, 1998):*

$$X = (W^TW)^{-1}W^TY \tag{98}$$
$$W^{(n)} = YX^T(XX^T)^{-1} \textit{ where } Y \textit{ is defined such that } S_N = YY^T. \tag{99}$$

*Proof.* The proof is direct. □

# C Derivation of the online EM algorithm for factor analysis

In this section we derive the online EM algorithm (Cappé and Moulines, 2009) for the factor analysis problem to compare it to our Recursive EM algorithm. We use the same notation than in the previous section where $v_1, \dots, v_N$ are our $N$ observations in dimension $d$ and $z_1, \dots, z_N$ are our latent variables in dimension $p$. Moreover, we suppose that each couple of variables $(z_t, v_t)$ are independent and belong to an exponential family given by $\log p(z_t, v_t|\theta) =< S(z_t, v_t), \phi(\theta) > -F(\theta)$, where $F$ is the log partition function, $\phi$ is a function which map the natural parameter and $S(z_t, v_t)$ are the sufficient statistics.

The online EM algorithm (Cappé and Moulines, 2009) consider the following fixed point equation with respect to the sufficient statistics $S$:

$$S = \mathbb{E}_{v_t}\mathbb{E}_{z_t\sim p(z|v_t,\theta^*(S))}[S(z_t, v_t)] = T(S) \tag{100}$$

$$\text{where } \theta^*(S) = \arg\max_{\theta\in\Theta} \sum_{t=1}^{N} < S(z_t, v_t), \phi(\theta) > -F(\theta), \tag{101}$$

and solve $T(S) - S$ using a stochastic root solver based on the Robbins–Monro algorithm (Robbins and Monro, 1951). The sufficient statistics $S$ and the optimal parameter $\theta$ are update online with an adaptive step $\gamma_t$ at each new incoming observations $v_t$ as follows:

$$S_t = (1 - \gamma_t)S_{t-1} + \gamma_t\mathbb{E}_{z_t\sim p(z|v_t,\theta_{t-1})}[S(z_t, v_t)] \quad \text{(E-step)} \tag{102}$$

$$\theta_t = \arg\max_{\theta\in\Theta} < S_t, \phi(\theta) > -F(\theta) \quad \text{(M-step)}. \tag{103}$$

In the case of factor analysis, the joint distribution is:

$$\log p(v_t, z_t) = -\frac{1}{2}\mathrm{Tr}[\Sigma^{-1}S(v_t, z_t)] - \frac{1}{2}\log\det\Sigma + c, \tag{104}$$

where the joint covariance is :

$$\Sigma = \begin{pmatrix} WW^T + \Psi & W \\ W^T & I_p \end{pmatrix}, \tag{105}$$

and the sufficient statistics are :

$$S(v_t, z_t) = \begin{pmatrix} v_t \\ z_t \end{pmatrix}\begin{pmatrix} v_t \\ z_t \end{pmatrix}^T = \begin{pmatrix} v_t v_t^T & v_t z_t^T \\ z_t v_t^T & z_t z_t^T \end{pmatrix}. \tag{106}$$

The expectation of the sufficient statistics gives

$$\mathbb{E}_{z_t\sim p(z|v_t,\theta_{t-1})}[S(v_t, z_t)] = \begin{pmatrix} v_t v_t^T & v_t\mathbb{E}[z_t^T|v_t] \\ \mathbb{E}[z_t|v_t]v_t^T & \mathbb{E}[z_t z_t^T|v_t] \end{pmatrix} \tag{107}$$

Rather than update the full matrix $S(v_t, z_t)$ we will update the blocks: $S_{1.t} = v_t v_t^T$, $S_{2.t} = \mathbb{E}[z_t|v_t]v_t^T$ and $S_{3.t} = \mathbb{E}[z_t z_t^T|v_t]$, which are necessary to compute the M-step.

Finally, using the same notation than in the previous Section, the online EM updates for factor analysis become:

$$
\begin{aligned}
&\textbf{Online EM}\\
&\textbf{E-step}\\
S_{1.t} &= (1-\gamma_t)S_{1.t-1} + \gamma_t v_t v_t^T\\
S_{2.t} &= (1-\gamma_t)S_{2.t-1} + \gamma_t \mathbb{E}[z_t|v_t]v_t^T\\
&= (1-\gamma_t)\mu_{t-1} + \gamma_t M_{t-1}^{-1} W_{t-1}^T \Psi_{t-1}^{-1} v_t v_t^T\\
S_{3.t} &= (1-\gamma_t)S_{3.t-1} + \gamma_t \mathbb{E}[z_t z_t^T|v_t]\\
&= (1-\gamma_t)S_{3.t-1} + \gamma_t (M_{t-1}^{-1} + M_{t-1}^{-1} W_{t-1}^T \Psi_{t-1}^{-1} v_t v_t^T \Psi_{t-1}^{-1} W_{t-1} M_{t-1}^{-1})\\
&\textbf{M-step}\\
W_t &= S_{2.t}^T S_{3.t}^{-1}\\
\Psi_t &= \mathrm{diag}(\mathrm{S_{1.t}}) - \mathrm{diag}(\mathrm{W_t S_{2.t}}).
\end{aligned}
\tag{108}
$$

To develop a limited memory version of this algorithm, we store only the diagonal of the high dimensional squared matrix $S_{1.t}$ and update it as follows:

$$
\mathrm{diag}(\mathrm{S_{1.t}}) = (1-\gamma_t)\mathrm{diag}(\mathrm{S_{1.t-1}}) + \gamma_t \mathrm{u_t} * \mathrm{u_t}, \tag{109}
$$

where $x * y$ is a component wise operation giving $x * y = \mathrm{diag}(\mathrm{xy^T})$ for two vectors.

To choose the step size $\gamma_t$ we must satisfy the Robins Monro rules:

$$
\sum_{t=1}^{N} \gamma_t = \infty \quad \sum_{t=1}^{N} \gamma_t^2 < \infty. \tag{110}
$$

We consider the following step recommended by Cappé and Moulines (2009):

$$
\gamma_0 = 1 \tag{111}
$$

$$
\gamma_t = \frac{1}{t^{0.6}} \quad \forall t > 0. \tag{112}
$$

Finally, in post-processing, we use a Polyak-Ruppert halfway averaging to improve the convergence as recommended by Cappé and Moulines (2009):

$$
\forall t > N/2 \quad st \quad \tilde{t} = t - N/2 > 0 \quad do:
$$

$$
\bar{W}_t = \frac{\tilde{t}-1}{\tilde{t}} W_{t-1} + \frac{1}{\tilde{t}} W_t \tag{113}
$$

$$
\bar{\Psi}_t = \frac{\tilde{t}-1}{\tilde{t}} \Psi_{t-1} + \frac{1}{\tilde{t}} \Psi_t. \tag{114}
$$

# D The outer product approximation in the general case.

We show in this Section the relation between the generalized Gauss Newton approximation and the Fisher matrix. This is a synthesis of previous results proposed by Ollivier (2018), Martens (2014) or again Cappé and Moulines (2009). The proof proposed here is a shorter version of the one developed by Ollivier (2018) in the Appendix $A$ of his article:

*Proof.* Let's $p$ be an exponential family such that:

$$p(y|\theta) = m(y_t)\exp(\eta^T T(y) - A(\eta)), \tag{115}$$

where $T(y)$ is the sufficient statistics, $A$ is the log partition function which satisfies $\nabla^2_\eta A(\eta) = Cov(y|\theta)$ and $\nabla_\eta A(\eta) = m = \mathbb{E}[(y|\theta)]$ and finally $\eta$ is the natural parameter which depends on $\theta$ through a function $h$, ie $\eta = h(\theta)$. Using twice the chain rules, the second derivative of the negative likelihood of $p$ writes :

$$-\frac{\partial^2 \ln p(y|\theta)}{\partial\theta^2} = -\frac{\partial h}{\partial\theta}\frac{\partial^2 \ln p(y|\theta)}{\partial h^2}\frac{\partial h}{\partial\theta}^T - \frac{\partial^2 h}{\partial\theta^2}\frac{\partial \ln p(y|\theta)}{\partial h} \tag{116}$$

$$= \frac{\partial h}{\partial\theta}Cov(y|\theta)\frac{\partial h}{\partial\theta}^T - \frac{\partial^2 h}{\partial\theta^2}(T(y) - m). \tag{117}$$

Taking the expectation under $y$ both sides, we obtain directly the relation:

$$\mathbb{E}_y[-\frac{\partial^2 \ln p(y/\eta)}{\partial\theta^2}] = \mathbb{F}(\theta) = \frac{\partial\eta}{\partial\theta}Cov(y|\theta)\frac{\partial\eta}{\partial\theta}^T. \tag{118}$$

And finally:

$$\mathbb{E}_\theta[\mathbb{F}(\theta)] = \mathbb{E}_\theta[\frac{\partial h}{\partial\theta}\mathrm{Cov}(y|\theta)\frac{\partial h}{\partial\theta}^T]. \tag{119}$$

Now for a generalized linear model such that $h(\theta) = x_t^T\theta$ the GGN approximation is exact since $\frac{\partial^2 h}{\partial\theta^2} = 0$, this completes the proof. □

# E Mirror prox

In this section we show that the iterated scheme defined equation (36) is equivalent to a Mirror prox update (Nemirovski, 2005). We recall first the connexion between the recursive variational scheme and the Mirror descent using the results initially derived by Khan et al. (2018) for the batch variational approach and extended by Lambert et al. (2021) for the recursive variational approach.

Considering an exponential family $q_\eta$ of natural parameter $\eta$, mean parameter $m$ and a strictly convex log partition function $F$ such that $q_\eta(\theta) = h(\theta)\exp(<\eta,\theta> - F(\eta))$, the recursive variational approximation problem between a target distribution $q_\eta$ and the one-sample posterior $p(\theta|y_t) \propto p(y_t|\theta)q_{\eta_{t-1}}(\theta)$ writes:

$$\arg\min_{\eta_t} \quad KL(q_{\eta_t}(\theta)|p(\theta|y_t)) \tag{120}$$

$$=\arg\min_{\eta_t} \quad \mathbb{E}_{q_{\eta_t}}[-\log p(y_t|\theta)] + B_F(\eta_{t-1},\eta_t), \tag{121}$$

where $B_F$ is the Bregman divergence associated to the strictly convex log partition function $F$. The critical point must satisfy:

$$\nabla_{\eta_t}\mathbb{E}_{q_{\eta_t}}[-\log p(y_t|\theta)] + (\eta_t - \eta_{t-1})\nabla^2 F(\eta_t) = 0, \tag{122}$$

which gives the following implicit fixed point equation on the natural parameter:

$$\eta_t = \eta_{t-1} + (\nabla^2 F(\eta_t))^{-1}\nabla_\eta\mathbb{E}_{q_\eta}[\log p(y_t|\theta)](\eta_t) \tag{123}$$

$$= \eta_{t-1} + \nabla_m\mathbb{E}_{q_m}[\log p(y_t|\theta)](m_t). \tag{124}$$

If we consider the function $f(m) = \mathbb{E}_{q_m}[-\log p(y_t|\theta)]$ and use the relation $\eta = \nabla F^*(m)$, this update can be interpreted as an implicit version of a Mirror descent on $f$ with step size one:

$$\nabla F^*(m_t) = \nabla F^*(m_{t-1}) - \nabla_m f(m_t) \text{ (dual descent with step 1)} \tag{125}$$
$$m_t = \nabla F(\eta_t). \text{ ( projection)} \tag{126}$$

This implicit scheme can be approximated using the Mirror prox algorithm (Nemirovski, 2005) with a step size one:

**Mirror prox**

$$\hat{\eta}_t = \nabla F^*(\hat{m}_t) = \nabla F^*(m_{t-1}) - \nabla_m f(m_{t-1})$$
$$\hat{m}_t = \nabla F(\hat{\eta}_t)$$
$$\eta_t = \nabla F^*(m_t) = \nabla F^*(m_{t-1}) - \nabla_m f(\hat{m}_t)$$
$$m_t = \nabla F(\eta_t).$$

In fact we have rather derived here a stochastic version of Mirror prox. Stochastic version of Mirror prox may inherit the good convergence properties of the original batch version if the function $f$ is convex and the step is adaptive (Bach and Levy, 2019). Here the setting is different: our function $f$ is not jointly convex in $\mu$ and $P$ and we consider a constant step of size one. However we have founded the stochastic version behave empirically well.

In the case where $q$ is a multivariate Gaussian distribution, the mean and natural parameters are given by $\eta = \nabla F^*(m) = \begin{pmatrix} \eta_1 = P^{-1}\mu \\ \eta_2 = -\frac{1}{2}P^{-1} \end{pmatrix}$ and $m = \nabla F(\eta) = \begin{pmatrix} m_1 = \mu \\ m_2 = P + \mu\mu^T \end{pmatrix}$.

The gradient with respect to the mean parameters $m_1, m_2$ can be expressed as the gradient with respect to the sources parameters $\mu, P$ using the chain rule:

$$\frac{\partial f}{\partial m_1} = \frac{\partial f}{\partial \mu}\frac{\partial \mu}{\partial m_1} + \frac{\partial f}{\partial P}\frac{\partial P}{\partial m_1} = \frac{\partial f}{\partial \mu} - 2\frac{\partial f}{\partial P}\mu \tag{127}$$
$$\frac{\partial f}{\partial m_2} = \frac{\partial f}{\partial \mu}\frac{\partial \mu}{\partial m_2} + \frac{\partial f}{\partial P}\frac{\partial P}{\partial m_2} = \frac{\partial f}{\partial P} \tag{128}$$

A step of the Mirror prox update:

$$\nabla F^*(m_t) = \nabla F^*(m_{t-1}) - \nabla_m f(m_{t-1}) \tag{129}$$
$$\iff \eta_t = \eta_{t-1} - \nabla_m f(m_{t-1}), \tag{130}$$

become, if we write $f(m(\mu, P)) = \mathbb{E}_{\theta\sim\mathcal{N}(\mu,P)}[\log p(y_t|\theta)]$:

$$\begin{pmatrix} P_t^{-1}\mu_t \\ -\frac{1}{2}P_t^{-1} \end{pmatrix} = \begin{pmatrix} P_{t-1}^{-1}\mu_{t-1} \\ -\frac{1}{2}P_{t-1}^{-1} \end{pmatrix} + \begin{pmatrix} \frac{\partial f}{\partial \mu}|_{\mu_{t-1},P_{t-1}} - 2\frac{\partial f}{\partial P}|_{\mu_{t-1},P_{t-1}}\mu_{t-1} \\ \frac{\partial f}{\partial P}|_{\mu_{t-1},P_{t-1}} \end{pmatrix} \tag{131}$$
$$\iff \begin{pmatrix} P_t^{-1}\mu_t \\ -\frac{1}{2}P_t^{-1} \end{pmatrix} = \begin{pmatrix} (P_{t-1}^{-1} - 2\frac{\partial f}{\partial P}|_{\mu_{t-1},P_{t-1}}\mu_{t-1} \\ -\frac{1}{2}P_{t-1}^{-1} \end{pmatrix} + \begin{pmatrix} \frac{\partial f}{\partial \mu}|_{\mu_{t-1},P_{t-1}} \\ \frac{\partial f}{\partial P}|_{\mu_{t-1},P_{t-1}} \end{pmatrix} \tag{132}$$
$$\iff \begin{pmatrix} P_t^{-1}\mu_t \\ -\frac{1}{2}P_t^{-1} \end{pmatrix} = \begin{pmatrix} P_t^{-1}\mu_{t-1} \\ -\frac{1}{2}P_{t-1}^{-1} \end{pmatrix} + \begin{pmatrix} \frac{\partial f}{\partial \mu}|_{\mu_{t-1},P_{t-1}} \\ \frac{\partial f}{\partial P}|_{\mu_{t-1},P_{t-1}} \end{pmatrix} \tag{133}$$
$$\iff \begin{pmatrix} P_t^{-1}\mu_t \\ -\frac{1}{2}P_t^{-1} \end{pmatrix} = \begin{pmatrix} P_t^{-1}\mu_{t-1} \\ -\frac{1}{2}P_{t-1}^{-1} \end{pmatrix} + \begin{pmatrix} -\mathbb{E}_{\theta\sim\mathcal{N}(\mu_{t-1},P_{t-1})}[\nabla_\theta \log p(y_t|\theta)] \\ \frac{1}{2}\mathbb{E}_{\theta\sim\mathcal{N}(\mu_{t-1},P_{t-1})}[\nabla^2_\theta \log p(y_t|\theta)] \end{pmatrix}. \tag{134}$$

The last derivation (134) comes from the Bonnet & Price formulas (Lin et al., 2019):

$$\nabla_\mu \mathcal{N}(\theta|\mu, P) = -\nabla_\theta \mathcal{N}(\theta|\mu, P) \tag{135}$$

$$\nabla_P \mathcal{N}(\theta|\mu, P) = \frac{1}{2}\nabla^2_\theta \mathcal{N}(\theta|\mu, P). \tag{136}$$

Rearranging terms and applying two times the update as in Mirror prox gives the iterated scheme defined in equation (36):

$$\hat{\mathbf{P}}_\mathbf{t}^{-1} = P_{t-1}^{-1} - \mathbb{E}_{\theta\sim\mathcal{N}(\mu_{t-1},P_{t-1})}[\nabla^2_\theta \log p(y_t|\theta)] \tag{137}$$

$$\hat{\mu}_\mathbf{t} = \mu_{t-1} + \hat{\mathbf{P}}_\mathbf{t}\mathbb{E}_{\theta\sim\mathcal{N}(\mu_{t-1},P_{t-1})}[\nabla_\theta \log p(y_t|\theta)] \tag{138}$$

$$\mathbf{P_t}^{-1} = P_{t-1}^{-1} - \mathbb{E}_{\theta\sim\mathcal{N}(\hat{\mu}_\mathbf{t},\hat{\mathbf{P}}_\mathbf{t})}[\nabla^2_\theta \log p(y_t|\theta)] \tag{139}$$

$$\mu_t = \mu_{t-1} + \mathbf{P_t}\mathbb{E}_{\theta\sim\mathcal{N}(\hat{\mu}_\mathbf{t},\hat{\mathbf{P}}_\mathbf{t})}[\nabla_\theta \log p(y_t|\theta)]. \tag{140}$$

If the expectations are replaced with a linearization around the last estimated, this updates are also equivalent to the extended iterated Kalman filter scheme (Jazwinski, 1970).

Applying the mirror-prox scheme to our logistic regression problem 5.3 without factor analysis, we see that the Gaussian well approximate the logistic posterior in figure 5. However, when we combine mirror-prox with factor analysis, the extra covariance update can make the mirror-prox scheme unstable. We have observed it is then preferable to skip the extra covariance update 139, i.e. using:

$$\hat{\mathbf{P}}_\mathbf{t}^{-1} = P_{t-1}^{-1} - \mathbb{E}_{\theta\sim\mathcal{N}(\mu_{t-1},P_{t-1})}[\nabla^2_\theta \log p(y_t|\theta)] \tag{141}$$

$$\hat{\mu}_\mathbf{t} = \mu_{t-1} + \hat{\mathbf{P}}_\mathbf{t}\mathbb{E}_{\theta\sim\mathcal{N}(\mu_{t-1},P_{t-1})}[\nabla_\theta \log p(y_t|\theta)] \tag{142}$$

$$\mu_t = \mu_{t-1} + \hat{\mathbf{P}}_\mathbf{t}\mathbb{E}_{\theta\sim\mathcal{N}(\hat{\mu}_\mathbf{t},\hat{\mathbf{P}}_\mathbf{t})}[\nabla_\theta \log p(y_t|\theta)]. \tag{143}$$

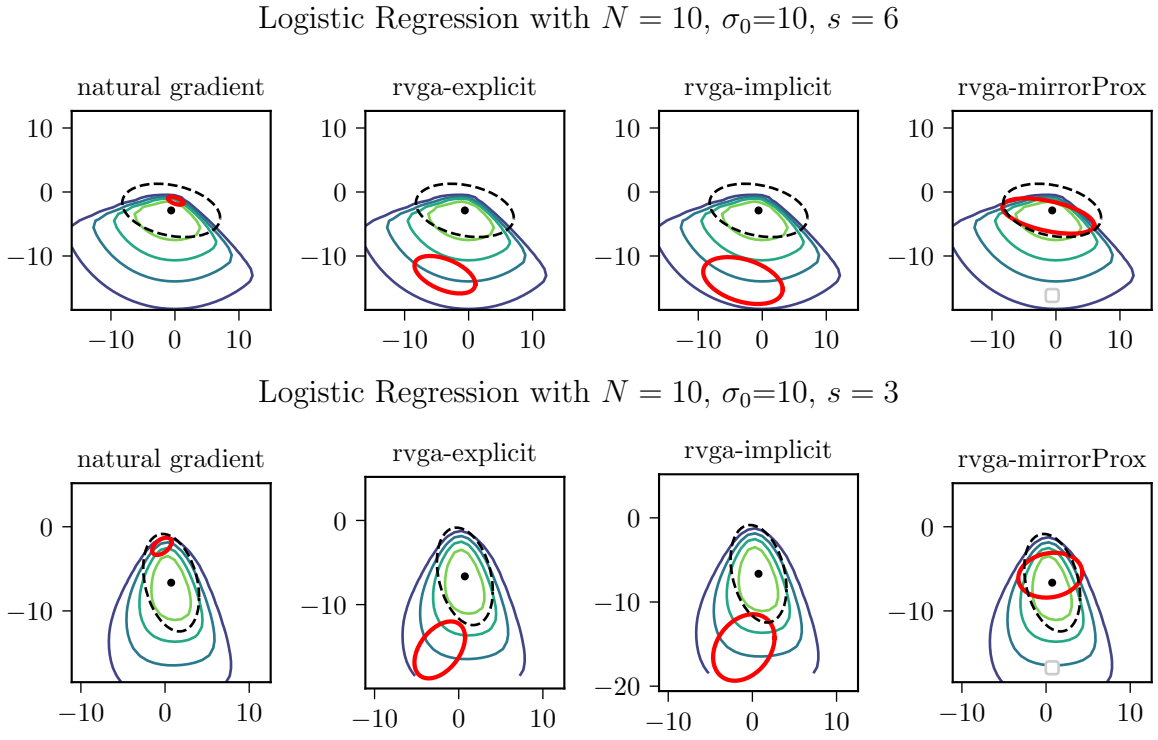


Figure 5: Gaussian approximation of the Bayesian logistic posterior with a sharp prior $\sigma_0 = 10$ for different algorithms. The confidence ellipsoids of the Gaussians at final time are shown in red. The contour lines of the true posterior are displayed in green. The batch Laplace ellipsoid is shown in dashed line. We compare the mirror-prox updates 137-140 (right column) with other variant of the updates : explicit using only 137-138 (second column), implicit using 54-55 (third column). The natural gradient (left column) correspond to a variant where the models are linearized. The mirror-prox schemes clearly better approximate the Bayesian logistic posterior.