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-h.lambert@intradef.gouv.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. It is also generalized to a large class of nonlinear models through a generalized outer product approximation using importance sampling and "Mirror Prox" to solve the recursive variational problem at each step.

1 Introduction

In machine learning and statistics, Bayesian inference aims at estimating the full posterior distribution of a parameter of interest θ, to better track the uncertainty of the learning process. Exact Bayesian inference is generally not tractable, and approximations are required. Variational approximation approach (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 θ given N observations $Y_N = y_1, \ldots, y_N$. The variational Gaussian approximation (Bar-

ber 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:

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

1.1 The recursive variational Gaussian approximation (RVGA)

While most methods developed so far to find the optimal parameters μ and P in the context of large scale problems revolve around stochastic descent algorithms (see for instance Ranganath et al. 2014), we opt for a recursive version of the variational Gaussian approximation problem in Equation (1), which leads to the RVGA updates (Lambert et al., 2021) we recall here. Let us suppose the observations $y_1, \ldots, y_N$ are independently distributed conditionally on θ. We let $y_t \sim p(.|\theta)$ where p is any differentiable probability distribution and θ is the unknown Bayesian parameter to be estimated. We want to estimate the posterior distribution $p(\theta|Y_N)$ using a single pass recursive variational approach where we constrain the approximated posterior q to be Gaussian, that is, of the form $\mathcal{N}(\theta|\mu_t, P_t)$ at each time step t, and we concisely denote this distribution by q_t.

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 RVGA updates—see (Lambert et al., 2021):

$$\mu_t = \mu_{t-1} - P_{t-1}\mathbf{E}_{\theta\sim\mathcal{N}(\mu_t,P_t)}[\nabla_\theta \ell_t(\theta)] \tag{3}$$

$$P_t^{-1} = P_{t-1}^{-1} + \mathbf{E}_{\theta\sim\mathcal{N}(\mu_t,P_t)}[\nabla^2_\theta \ell_t(\theta)], \tag{4}$$

where ℓ_t is the one sample loss $\ell_t(\theta) = -\log p(y_t|\theta, x_t)$, and x_t is the input associated to the observation or label y_t. Theses updates resemble an averaged version of the online Newton algorithm and provide a new stochastic second-order descent in the sense that it uses an online approximation of the Hessian matrix. Because μ_t and P_t appear both in the left and right hand sides of the equations above, this only provides an implicit scheme equivalent to the approximate optimization problem we seek to solve. As long as it can be resolved, it spares us the use of stochastic optimization methods that involve step size tuning such as the stochastic natural descent proposed by Khan et al. (2018). The implicit scheme above was solved for the linear and logistic regression problems leading to an algorithm which is more stable than the natural gradient algorithm (see Lambert et al. 2021 for more details). If this algorithm seamlessly scales with the number of observations N, it has quadratic cost in the parameter dimension d, assuming we avoid direct computation of the inverse of P_t^{-1} using the Woodbury formula, hindering its use in contexts involving high dimension. The objective of this paper is to provide the user with an approximate version that scales linearly in d both in terms of computation and storage costs.

1.2 The linear regression case and the recursive EM for factor analysis

We consider first the simple case of linear regression, that is, outputs are produced by a linear model $y_t = x_t^T\theta + w_t$ where $w_t \sim \mathcal{N}(0, \mathbf{I})$, and the prior is Gaussian $\theta_0 \sim \mathcal{N}(0, P_0)$. The update (4) then boils down to updates in information form:

$$P_t^{-1} = P_{t-1}^{-1} + x_t x_t^T. \tag{5}$$

When the parameter dimension d is very large, it may be impossible to store the $d \times d$ covariance matrix P_t, and the update (5) may become intractable. This prompts the use of an approximation of the inverse covariance matrix (precision matrix) that may be stored using a reduced number of parameters.

Iterating equation (5) from $t = 1$ to $t = N$, we see that the covariance matrix P_N of the Bayesian parameter θ after N observations is related to the empirical covariance of the inputs $\frac{1}{N}\sum_{i=1}^N x_i x_i^T$. This suggest to use the factor analysis (FA) structure (Harman, 1967, Tipping and Bishop, 1999), initially proposed to approximate empirical covariance matrix as a means to approximate P_t^{-1}.

The factor analysis (FA) approximation to the precision matrix takes the form of a "low rank + diagonal" matrix, that is,

$$P^{-1} \underset{FA}{\approx} WW^T + \Psi \tag{6}$$
$$\text{with } W \in \mathcal{M}(d \times p) \text{ and } \Psi \text{ a diagonal matrix.}$$

In this way the $d \times d$ precision matrix is encoded in $d(p+1)$ parameters, where we let $p \ll d$, allowing it to be stored even for large d. In FA, the columns of W are interpreted as the reduced number p of factors that underpin the data, and Ψ as the covariance matrix of independent noises affecting observations. Our FA estimation problem then amounts to finding the matrices W, Ψ that best fit the precision matrix. Since this precision matrix may be interpreted as the sample covariance matrix of the inputs, this problem is a maximum likelihood problem which may thus be addressed via the usual expectation maximization (EM) algorithm (Dempster et al., 1977).

To maintain an online and limited memory algorithm we cannot use the batch EM algorithm. We may rather use the online EM algorithm from Cappé and Moulines (2009) which processes inputs sequentially and gives state-of-the-art results. As an alternative to the online EM, though, we instead introduce a new recursive version of the EM algorithm which suits well for our factor analysis problem and which is derived from a fixed point equation on latent factors computed in a closed form. Our algorithm is parameter free, contrary to the online EM, and converges faster to the latent factors W, Ψ after only one pass through the data. This allows for a parameter-free algorithm with storage cost being linear in d and overall computation cost $O(Nd)$, yielding an approximate posterior distribution $\mathcal{N}(\mu_N, P_N)$ for the parameter θ given all the observations $y_1, \ldots, y_N$ after a single pass over the data.

1.3 The limited-memory RVGA

If we now turn to the general case, that is, beyond linear regression, the evolution of the precision matrix at each step is given by the general RVGA updates (3)-(4). If we have approximated the precision matrix with a factor analysis at step $t-1$, i.e., $P_{t-1}^{-1} \underset{FA}{\approx} W_{t-1}W_{t-1}^T + \Psi_{t-1}$, we then seek to approximate the RVGA updates (3)-(4) at step t as:

$$P_t^{-1} = W_{t-1}W_{t-1}^T + \Psi_{t-1} - \mathbf{E}_{\theta\sim\mathcal{N}(\mu_t,P_t)}[\nabla_\theta^2 \log p(y_t|\theta, x_t)] \approx W_t W_t^T + \Psi_t \tag{7}$$

$$\mu_t = \mu_{t-1} + (W_{t-1}W_{t-1}^T + \Psi_{t-1})^{-1}\mathbf{E}_{\theta\sim\mathcal{N}(\mu_t,P_t)}[\nabla_\theta \log p(y_t|\theta, x_t)]. \tag{8}$$

Albeit solvable in the logistic regression case (Lambert et al., 2021), the above scheme is difficult to solve in the general case because it is implicit, involves intractable expectations and uses a Hessian matrix which may not fit in memory in high dimension. We propose the following solutions to tackle these difficulties:

1. We apply "Mirror Prox" (Nemirovski, 2005), a.k.a., the extra-gradient method, to approximate the implicit scheme above.

2. We propose a new importance sampling method to sample from a Gaussian distribution with a structured precision matrix $P_t^{-1} = W_t W_t^T + \Psi_t$. This method allows us to approximate expectations of the type $\mathbf{E}_{\theta\sim\mathcal{N}(\mu_t,P_t)}$ without storing a $d \times d$ matrix.

3. We use the generalized Gauss-Newton approximation (Martens, 2014) to approximate the Hessian matrix $\nabla^2_\theta \log p(y_t|\theta)$ in an outer product form and apply our new recursive EM algorithm to it to meet the linear memory requirements in the parameter dimension d.

1.4 Organization of the paper

The paper is organized as follows: in Section 2 we discuss how our approach differs from several related studies on large-scale second-order estimation procedures. In Section 3, we model our problem as a two-stage variational approximation scheme and show how the second stage can be interpreted as a maximum likelihood problem for factor analysis. In Section 4, we introduce our new recursive EM algorithm for factor analysis and use it to approximate large-scale covariance matrices with a memory requirement linear in d. In Section 5 we apply this recursive EM to the originally considered Bayesian inference problem, with applications to high-dimensional linear and logistic regressions for which the RVGA updates are available in a closed form. In Section 6 we extend the application to any probability model using Mirror Prox to approximate the implicit scheme and the generalized Gauss-Newton approximation to apply our recursive EM algorithm. We introduce also a novel method to sample from a Gaussian whose dispersion is encoded in a precision matrix structured with a factor analysis model. In Section 7 our algorithm is evaluated on synthetic data for large scale matrix approximation, and for linear and logistic regression. We evaluate Mirror Prox and our sampling method on logistic regression problems for which we can compute updates in closed form thus providing a baseline for comparison.

2 Related work

Several second-order algorithms have been proposed which scale to both large sample sizes and large dimensions. These algorithms have been used in two main areas of applications: large scale machine learning and Kalman filtering in high dimension. These two applications are strongly related; indeed the extended Kalman filter with a static state is equivalent to the online natural gradient under mild conditions, as shown by Ollivier (2018). Moreover, both fall within the variational inference framework where they happen to be a particular case of our recursive variational Gaussian approximation (see Lambert et al. 2021 for proof details). A deeper connection between the communities of stochastic optimization and Kalman filtering may provide innovative approaches to tackle high dimension nonlinear problems in a stochastic way. We now review a few solutions proposed in each of these areas to solve high dimensional problems.

2.1 Large scale machine learning with second-order stochastic optimization

The first type of application concerns accelerating the convergence of stochastic gradient descent in large scale machine learning typically to train deep neural networks. The second-order information is used to adapt the step-size to each coordinate of the gradient depending on the local curvature. Multiple passes on the data are then needed to make the descent converge and the second-order information is not generally used to quantify the uncertainty on the learned parameters. Theses algorithms can use severe approximations of the Hessian matrix, like Adagrad (Duchi et al., 2011) which keeps only the diagonal terms of the outer-product approximation of the Hessian. Lecun et al. (1993) proposed a "recipe" to automatically compute the learning rate of the stochastic gradient descent where the main eigenvalues of the Hessian matrix are computed implicitly with a power method. Puskorius and Feldkamp (1991) and Roux et al. (2008) used the structure of the nonlinear function, a neural network, to approximate the Hessian matrix with a block-diagonal matrix. They then interpret this matrix as the covariance matrix of the parameters.

Finally, in classical optimization, Newton method does not scale well for large dimension. Quasi-Newton algorithms approximate the Hessian using only evaluations of the gradient. L-BFGS (Liu and Nocedal, 1989) is a limited memory version of the quasi-Newton descent which has much been used for optimization in high dimension and has been recently proposed in a stochastic version (Byrd et al., 2015). A proximal variant has also been proposed, considering low-rank structures of the form "symmetric+/-low-rank" (Becker et al., 2019).

2.2 Kalman filtering with covariance estimation

In Kalman filtering, we want to estimate, in real time, the state (or the latent variable) of a dynamical system. Kalman filtering has numerous applications in aerospace engineering, robotics and signal processing, see, e.g., Barrau and Bonnabel 2016 and Grewal and Andrews 2010 for recent contributions. The extended Kalman filter (EKF) is widely used in those fields to estimate the covariance of the parameters online using successive linearization of the observation and dynamic models. These approximations can make the filter diverge and improving the stability of the EKF is an active research domain (Barrau and Bonnabel, 2018, Arasaratnam and Haykin, 2009). Kalman filters are online in essence and scale to large amounts of observations: a single pass on the observations is sufficient in the filtering mode to estimate the covariance of the estimation and two passes, namely one forward and one backward, are used in the smoothing setting. Different variants have been proposed to tackle high dimensions: the ensemble filter (Evensen, 1994) approximates the covariance matrix with sampling using a Monte Carlo approach and has been widely used in data assimilation for weather forecasting and oceanography. In the same field of application, the SEEK filter (Pham et al., 1998) is based on a singular value decomposition (SVD) of the covariance matrix and provides a low rank Riccati update. Unfortunately, this factorization does not provides an invertible covariance matrix. In robotics, Thrun et al. (2004) have proposed the sparse extended information filter (SEIF) Kalman filter to process large maps in SLAM (simultaneous localization and mapping) problems. The idea is to use a sparse approximation of the information (i.e., precision) matrix that matches the fact that conditionally on the robot's state features, uncertainty on the features that compose the map is decorrelated.

2.3 Online variational inference

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.

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 a 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".

3 Two-stage variational approximation

In the original recursive variational problem in Equation (2), we search, at each new observation y_t, a target distribution $q_t = \mathcal{N}(\theta|\mu_t, P_t)$ with $\theta \in \mathbb{R}^d$. To limit memory requirements, we want to constrain the successive covariance matrices P_t to be writable as $P_t^{-1} = W_t W_t^T + \Psi_t$ where W_t is a $d \times p$ matrix and Ψ is a $d \times d$ diagonal matrix. A first approach is to address the following direct optimization problem:

$$q_t(\theta) = \underset{\mu, W, \Psi}{\arg\min} \quad KL\left(\mathcal{N}(\theta|\mu, (WW^T + \Psi)^{-1}) \| c_t p(y_t|\theta) q_{t-1}(\theta)\right), \tag{9}$$

where we recall that we use the Bayes formula $p(\theta|y_t) = c_t p(y_t|\theta) q_{t-1}(\theta)$. Problem (9) has been tackled by Ong et al. (2018) where the authors used a stochastic gradient algorithm to compute the optimal parameters. As an alternative, to meet the desired constraint of "low-rank plus diagonal" precision matrix, we advocate a different approach that has the advantage of being parameter-free, that is, where no step sizes need to be tuned. It is based on a two-stage variational approximation that consists of 1) a projection of the posterior distribution $p(\theta|y_t)$ onto the space of Gaussian distributions following the classical variational inference approach, and 2) a projection of the obtained Gaussian distribution onto the set of structured Gaussian distributions with a precision matrix

factorized as "low-rank plus diagonal". Mathematically, this writes:

Two-stage variational approximation

$$q_t^*(\theta) = \mathcal{N}(\theta|\mu_t^*, P_t^*) = \arg\min_{\mu,P} \quad KL(\mathcal{N}(\theta|\mu, P)\|c_t p(y_t|\theta) q_{t-1}(\theta)) \tag{10}$$

$$q_t(\theta) = \mathcal{N}(\theta|\mu_t^*, \tilde{P}_t) = \arg\min_{W,\Psi} \quad KL(\mathcal{N}(\theta|\mu_t^*, (WW^T + \Psi)^{-1})\|q_t^*(\theta)). \tag{11}$$

These are known as information projections or I-projections in information geometry. The first projection is motivated by the necessity to restrict the search of posterior distributions to a "nice" family of distributions with easily interpretable parameters, while the second projection is performed to keep a memory cost being linear in the dimension d. The first point is the object of our previous work (Lambert et al., 2021), whereas the second one and more generally the proposed two-stage approach are the object of the present paper, dedicated to the application of RVGA to large dimensional problems.

In the linear case, though, these two projections are equivalent to the direct projection of the posterior distribution onto the set of "low-rank plus diagonal" matrices, as the first projection is trivial. Indeed, $c_t p(y_t|\theta) q_{t-1}(\theta)$ is then a Gaussian, so that it is equal to $q_t^*(\theta)$, see (Lambert et al., 2021).

Proposition 1. *Assume that the observations are produced by a linear Gaussian model with Gaussian prior. Then, the two-stage approach* (10)-(11) *directly solves the one-stage minimization problem* (9).

Let us now come back to the general case, where our two-stage approach does not solve the more direct optimization problem (9). The first projection (10) leads to the RVGA update equations given by (see Lambert et al. 2021 for the detailed proof):

$$\mu_t^* = \mu_{t-1} + P_{t-1} \mathbf{E}_{\theta\sim\mathcal{N}(\mu_t^*, P_t^*)}[\nabla_\theta \log p(y_t|\theta)] \tag{12}$$

$$P_t^{*-1} = P_{t-1}^{-1} - \mathbf{E}_{\theta\sim\mathcal{N}(\mu_t^*, P_t^*)}[\nabla_\theta^2 \log p(y_t|\theta)]. \tag{13}$$

The second projection (11) is a divergence between two Gaussians having identical means. It turns out that the divergence (11) may be algebraically related to a maximum likelihood problem. Indeed we have:

$$\min_{W,\Psi} \quad KL(\mathcal{N}(\theta|\mu_t^*, (WW^T + \Psi)^{-1})\|\mathcal{N}(\theta|\mu_t^*, P_t^*)) \tag{14}$$

$$= \min_{W,\Psi} \quad KL(\mathcal{N}(\theta|0, P_t^{*-1})\|\mathcal{N}(\theta|0, WW^T + \Psi)), \tag{15}$$

where the last equation comes from changing covariance P to precision P^{-1} as

$$KL(\mathcal{N}(\mu, P_1)\|\mathcal{N}(\mu, P_2))) = \frac{1}{2}(\mathrm{Tr} P_2^{-1} P_1 + \log \frac{|P_2|}{|P_1|}) - \frac{d}{2} = KL(\mathcal{N}(0, P_2^{-1})\|\mathcal{N}(0, P_1^{-1})). \tag{16}$$

The parameters we seek W, Ψ appear now on the right side of the KL. If the left side distribution in Equation (15) is interpreted as an empirical distribution, minimizing the KL divergence with respect to the right side distribution in Equation (15) is equivalent to maximizing a likelihood on observations having as empirical covariance P_t^{*-1} (see Remark 1 below for more details). Moreover, the left side distribution can be interpreted as the distribution of the gradients $u = \nabla_\theta \log q_t^*(\theta) = -P_t^{*-1}(\theta - \mu_t^*)$ considering the relation:

$$\mathrm{E}[u] = \mathrm{E}[\nabla_\theta \log q_t^*(\theta)] = 0, \qquad \mathrm{Cov}(u) = \mathrm{E}[uu^T] = P_t^{*-1}. \tag{17}$$

As a consequence, finding W and Ψ is equivalent to finding a factor analysis form (Harman, 1967) on the gradient:

$$u = \nabla_\theta \log q_t^*(\theta) = Wz + \varepsilon \text{ where } z \sim \mathcal{N}(0, I_p) \text{ and } \varepsilon \sim \mathcal{N}(0, \Psi). \tag{18}$$

This approach may be compared to the one developed by Roux et al. (2008) where the authors used a singular value decomposition on the incremental Gram matrix of the gradients.

Remark 1. *The problem of minimizing a KL divergence between an empirical distribution and a target distribution is equivalent to a maximum likelihood problem. This result comes from geometry of information (Nielsen, 2020). In our case, we can let the log likelihood appear explicitly from our variational problem through the Stein divergence. The right-side KL-divergence* (15) *can be rewritten as*

$$KL(\mathcal{N}(0, P^{*-1}) \| \mathcal{N}(0, WW^T + \Psi)) \tag{19}$$

$$= \frac{1}{2}\mathrm{Tr}(WW^T + \Psi)^{-1}P^{*-1} + \frac{1}{2}\log\det(WW^T + \Psi) - \frac{1}{2}\log\det(P^{*-1}) - \frac{d}{2} \tag{20}$$

$$= \frac{1}{2}D_S(P^{*-1} \| WW^T + \Psi), \tag{21}$$

where D_S is the Stein Divergence between covariance matrices defined by $D_S(X\|Y) = \mathrm{Tr}(XY^{-1}) - \log\det(XY^{-1}) - d$. Moreover, minimizing this Stein divergence (21) *is equivalent to maximizing the* log *likelihood :*

$$\max_{W,\Psi} \quad \log \mathcal{N}(v_1, \ldots, v_K | 0, WW^T + \Psi) \tag{22}$$

$$= -\frac{K}{2}\mathrm{Tr}[(WW^T + \Psi)^{-1}\frac{1}{K}\sum_{i=1}^{K} v_i v_i^T] - \frac{K}{2}\log\det(WW^T + \Psi) - \frac{1}{2}\log(2\pi), \tag{23}$$

if the samples $v_1, \ldots, v_K$ have a null empirical mean and an empirical covariance matrix $\frac{1}{K}\sum_{i=1}^{K} v_i v_i^T = P^{*-1}$. Finally, if we introduce a latent variable z such that $v|z \sim \mathcal{N}(Wz, \Psi)$ we can use expectation-maximization algorithm (Dempster et al., 1977) to approximate the maximum likelihood and find the factor analysis parameters.

Our approach in a nutshell: to summarize, we have considered the problem of estimating the posterior $p(\theta \mid y_1, \ldots, y_N)$ recursively through variational approximation, allowing for large datasets to be handled, i.e., large N. The variational distribution q_t at current step incorporates the current observation y_t to the approximated posterior, and is assumed to be Gaussian with mean μ_t and covariance P_t. Moreover, to attack problems that also involve a high dimensional parameter θ, we pursue storage and computation costs being linear in the parameter dimension d. To this aim, we approximate the precision matrix at each step P_t^{-1} by a low-rank plus diagonal matrix $W_t W_t^T + \Psi_t$. This results in the two-stage variational scheme (10)-(11) where (11) is a low dimensional approximation to the original variational approximation (10). The problem (10) has been thoroughly discussed in our prior work, see Lambert et al. 2021 and the remainder of the present paper is devoted to the extension to high dimension, with a focus on problem (11). The latter problem may be interpreted as a projection of the precision matrix at each step P_t^{*-1} onto the set of low-rank plus diagonal matrices $W_t W_t^T + \Psi_t$ in the sense of the KL divergence. Finally, it turns out that solving this problem is equivalent to determining the maximum likelihood parameters of a factor analysis model with sample covariance matrix $S = P^{*-1}$. This allows for bringing maximum likelihood algorithms to bear for the original optimization problem (11).

4 Recursive factor analysis approximation

In this Section we develop our new recursive EM algorithm as a parameter-free alternative to the online EM algorithm (Cappé and Moulines, 2009), which is more general than the proposed following method, but may prove more difficult to apply in the present specific context of recursive factor analysis.

Suppose we have factorized at step $t-1$ the current covariance matrix P_{t-1} of our RVGA estimator as $(W_{t-1}W_{t-1}^T+\Psi_{t-1})^{-1}$. We thus seek at step t the closest matrix of the form $W_tW_t^T+\Psi_t$ in the sense of the KL divergence to the new precision matrix P_t^{-1} given by the RVGA update (4):

$$P_t^{-1} = W_{t-1}W_{t-1}^T + \Psi_{t-1} - \mathbf{E}_{\theta\sim\mathcal{N}(\mu_t,P_t)}[\nabla_\theta^2 \log p(y_t|\theta, x_t)]. \tag{24}$$

To apply our recursive EM algorithm we will suppose we can express Equation (24) in the following recursive outer-product form:

$$P_t^{-1} = W_{t-1}W_{t-1}^T + \Psi_{t-1} + u_tu_t^T. \tag{25}$$

The term u_t will match the input x_t in the linear regression case, but it will take a different form in the general nonlinear case where we will consider a generalized Gauss-Newton approximation of the Hessian matrix (see Section 5).

We develop the recursive EM algorithm in two steps. First, we consider the classical batch EM algorithm for covariance approximation $S_N = \frac{1}{N}\sum_{i=1}^N u_iu_i^T$ and write it through fixed point equations. Second, we apply this fixed point update to the recursive outer-product structure in a fashion that limits the storage and computational cost to $O(d)$ at each step.

4.1 Batch EM as a fixed point equation

In this section we consider N samples $u_1, \dots, u_N$ sampled from a centered Gaussian distribution. In the learning framework, these quantities may be related to the input features associated to the labels $y_1, \dots, y_N$ but in this section we will consider them independently as "samples".

The FA model, in its classical batch form, writes as follows

$$u = Wz + \varepsilon \quad \text{with } \varepsilon \sim \mathcal{N}(0, \Psi) \quad \text{and } z \sim \mathcal{N}(0, \mathbf{I}_p), \tag{26}$$

where W, Ψ are the parameters to be inferred. The log-likelihood of the samples $u_1, \dots, u_N$ writes

$$\log p(u_1, \dots, u_N \mid W, \Psi) = -\frac{N}{2}\log\det(W_tW_t + \Psi_t) - \frac{N}{2}\mathrm{Tr}[(W_tW_t + \Psi_t)^{-1}S_N] - \frac{1}{2}\log(2\pi), \tag{27}$$

with $S_N = \frac{1}{N}\sum_{i=1}^N u_iu_i^T$ the sample covariance matrix of data $u_1, \dots, u_N$. Maximizing this likelihood may be tackled through an EM algorithm (Dempster et al., 1977). At the expectation step, the latent variables z_t are estimated conditionally on the observation and the current parameter estimates. At the maximization step, the latent variables are assumed fixed and the parameters are adjusted to maximize likelihood. Let us apply it to the problem at hand, see also (Harman, 1967, Tipping and Bishop, 1999).

Gaussian conditioning formulas prove the posterior of the latent variable z_t writes

$$p(z_t|u_t, W, \Psi) = \mathcal{N}(M^{-1}W^T\Psi^{-1}u_t, M^{-1}) \text{ where } M = \mathbf{I}_p + W^T\Psi^{-1}W. \tag{28}$$

The total likelihood for the batch problem writes :

$$\log p(u_t, z_t|W_t, \Psi_t) = \sum_{t=1}^{N} \log p(u_t|z_t, W_t, \Psi_t) + \sum_{t=1}^{N} \log p(z_t) \tag{29}$$

$$= -\frac{N}{2}(u_t - W_t z_t)^T \Psi_t^{-1}(u_t - W_t z_t) - \frac{N}{2}\log\det\Psi_t - \frac{N}{2}\log(2\pi) + \sum_{t=1}^{N}\log p(z_t). \tag{30}$$

For the E-step, we compute the expected total likelihood $\mathbb{E}_{z_t|u_t,W_t,\Psi_t} \log p(u_t, z_t|W, \Psi)$. For the M-step, it is maximized with respect to W and Ψ. After some calculation recapped in Appendix A.1, the batch EM yields the following updates:

E-Step:

$$\begin{aligned}\mathbf{E}[z_t|u_t] &= M^{-1}W^T\Psi^{-1}u_t \\ \mathbf{E}[z_t z_t^T|u_t] &= M^{-1} + \mathbf{E}[z_t|u_t]\mathbf{E}[z_t|u_t]^T.\end{aligned} \tag{31}$$

M-Step:

$$\begin{aligned}W_{new} &= \sum_{t=1}^{N} u_t \mathbf{E}[z_t|u_t]^T \Big(\sum_{t=1}^{N}\mathbf{E}[z_t z_t^T|u_t]\Big)^{-1} \\ \Psi_{new} &= \mathrm{diag}\Big(S_N - W_{new}\frac{1}{N}\sum_{t=1}^{N}\mathbf{E}[z_t|u_t]u_t^T\Big).\end{aligned} \tag{32}$$

These batch updates can be reformulated as a fixed point equation as shown in the following lemma which extends the proof developed for the probabilistic principal component analysis (Tipping and Bishop, 1999) to the more general factor analysis case:

Lemma 1. *The batch expectation-maximization algorithm* (31)-(32) *to infer* W, Ψ *from an empirical covariance matrix* $S_N = \frac{1}{N}\sum_{t=1}^{N} u_t u_t^T$ *is equivalent to iterating on the following fixed point:*

Fixed-point equations for EM

$$\begin{aligned}W_n &= S_N\Psi^{-1}W(\mathbf{I}_p + M^{-1}W^T\Psi^{-1}S_N\Psi^{-1}W)^{-1}, \\ &\quad \textit{where } M = \mathbf{I}_p + W^T\Psi^{-1}W \\ \Psi &= \mathrm{diag}(S_N - W_n M^{-1}W^T\Psi^{-1}S_N) \\ W &= W_n.\end{aligned} \tag{33}$$

The proof is detailed in Appendix A.2. The solution of these fixed point equations are also the solution of the fixed point equations obtained when maximizing directly the (conditional) likelihood (27) (ML):

Fixed-point equations for ML

$$\begin{aligned}W &= S_N(WW^T + \Psi)^{-1}W \\ \Psi &= \mathrm{diag}(S_N - WW^T).\end{aligned} \tag{34}$$

See proof in Appendix B.2. The advantage of using EM fixed point iterations (33) rather than ML fixed point iterations (34) is that in the former we are guaranteed to increase the likelihood while in the latter not.

Remark 2. ***Connexion with the singular value decomposition (SVD)***
In the classical ML approach, the ML fixed point equation (34) *is generally not used directly and the update on W is rewritten using a SVD of S as recalled in Appendix C.1. The SLANG approach (Mishkin et al., 2018) uses also a SVD and is equivalent to the ML fixed point equation* (34) *where only one iteration is considered, neglecting the coupling between W and Ψ. It can be shown that the EM-fixed point updates* (33) *compute implicitly an SVD, see Appendix C.2. Indeed, considering the asymptotical probabilistic principal component analysis form (PPCA), i.e., $\Psi = \sigma I$, where we let the scalar parameter σ tend to* 0*, it turns out that the EM fixed point updates boil down to the EM PCA method (Roweis, 1998) whereas the ML fixed point update boils down to a power method (Von Mises and Pollaczek-Geiringer, 1929). In particular if W is a vector ($p = 1$), the ML fixed point gives $W \leftarrow S_N \frac{W}{||W||^2}$ and the EM fixed point gives $W \leftarrow S_N \frac{W(W^T S_N W)^{-1}}{||W||^2}$. Since the EM fixed points and the ML fixed point are the same, EM implicitly performs a power method but is more stable as shown by Roweis (1998).*

This result concerns only the batch factor analysis. To come back to our problem, we need at step t to approximate the matrix $W_{t-1}W_{t-1}^T + \Psi_{t-1} + u_t u_t^T$. This may be performed applying the fixed-point EM update above to $S_N = W_{t-1}W_{t-1}^T + \Psi_{t-1} + u_t u_t^T$ to compute W_t and Ψ_t. To avoid storing and manipulating $d \times d$ matrices, though, a limited memory recursive implementation is developed in Section 4.2.

4.2 The recursive EM algorithm for factor analysis

We are now ready to introduce the recursive EM algorithm as a version of the previous batch EM algorithm that avoids storing and manipulating $d \times d$ matrices, and which requires a number of operations that scales linearly with the dimension d. Our novel recursive EM method described in Algorithm 1 consists in successively performing a few cycles on the fixed-point equation (33) letting $S_N = W_{t-1}W_{t-1}^T + \Psi_{t-1} + u_t u_t^T$, where we develop and rearrange the terms to avoid any operations that require storing or multiplying $d \times d$ matrices. The derivation of this algorithm is detailed in Appendix A.3.

The operation "$x*y$" which appears at the last line is a memory cheap component wise operation of two matrices x and y of same size $d \times p$. If $p = 1$ we have $x * y = \mathrm{diag}(xy^T)$, if $p > 1$ we have $\sum_{i=1}^p x * y[., i] = \mathrm{diag}(xy^T)$. The parameter β controls the number of inner loops to make the EM algorithm converge at each step, although a fixed number of three inner loops may be sufficient in practice. The parameter ε need to be set at a small value. Its purpose is to avoid initializing on $W = 0$, which is also a stationary point of the fixed-point equation. The parameter σ_0^2 is the regularization parameter which spares the inversion of the matrix $WW^T + \Psi$ at the first iterations, and may be set to a value being of the order of magnitude of the averaged norm of the data $\frac{1}{N}\sum_{t=1}^N ||u_t||$.

The diagonal matrix Ψ 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/ψ^T where $/\psi^T$ operates on the rows of W^T.

Remark 3. *The samples u_t may be a vector or a sub-matrix $u_t = \frac{1}{\sqrt{K}}\left(c_1 \cdots c_K\right)$ composed of a mini-batch of K vectors $c_1, \ldots, c_K$ such that $u_t u_t^T = \frac{1}{K}\sum_{i=1}^K c_i c_i^T$. This mini-batch version will be particularly useful when we will consider the nonlinear case.*

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 parameter are finely tuned but our approach is

Algorithm 1: Recursive expectation-maximization algorithm for factor analysis

Result: W, Ψ
Given N inputs $u_1, \dots, u_N$;
$\Psi_0 = \sigma_0^2 \mathbf{I}_d$;
for $i \leftarrow 1$ **to** p **do**
  $W_0[i] = \mathcal{N}(0, \varepsilon \mathbf{I}_d)$;
end
for $t \leftarrow 1$ **to** N **do**
  $W_t = W_{t-1}$;
  $W_{new} = W_{t-1}$;
  $\Psi_t = \Psi_{t-1}$;
  repeat
    $W_t = W_{new}$;
    $M_t = \mathbf{I}_p + W_t^T \Psi_t^{-1} W_t$;
    $V_t = u_t(u_t^T \Psi_t^{-1} W_t) + W_{t-1}(W_{t-1}^T \Psi_t^{-1} W_t) + \Psi_{t-1} \Psi_t^{-1} W_t$;
    $W_{new} = V_t(\mathbf{I}_p + M_t^{-1} W_t^T \Psi_t^{-1} V_t)^{-1}$;
    $\Psi_t = u_t * u_t + \sum_{i=1}^p W_{t-1} * W_{t-1}[., i] + \Psi_{t-1} - \sum_{i=1}^p W_{new} M_t^{-1} * V_t[., i]$;
  until $||W_t - W_{new}|| < \beta$;
end

parameter-free and appears more natural in the context of factor analysis since it is derived directly from the fixed point equation. Moreover our experiments indicate it converges faster than the online EM and gives new state-of-the-art results for online EM applied to factor analysis (see Section 7.1).

5 Limited memory RVGA (L-RVGA) : linear and logistic regression case

To solve the variational optimization problem introduced in Section 3 we need to compute two Kullback-Leibler projections (10) and (11). The solution to the first projection must satisfy the RVGA updates (12)-(13) that we recall herein:

$$P_t^{-1} = P_{t-1}^{-1} - \mathbf{E}_{\theta \sim \mathcal{N}(\mu_t, P_t)}[\nabla_\theta^2 \log p(y_t|\theta, x_t)], \tag{35}$$

$$\mu_t = \mu_{t-1} + P_{t-1} \mathbf{E}_{\theta \sim \mathcal{N}(\mu_t, P_t)}[\nabla_\theta \log p(y_t|\theta, x_t)]. \tag{36}$$

In the previous section we have proposed an algorithm to solve the second projection based on a recursive EM factor analysis to approximate the large scale matrix P_t^{-1} returned by the standard RVGA update. We now combine these two solutions to solve the variational problem in a memory-efficient way. We consider first the linear regression case in Section 5.1 and extend these results to any generalized linear model (GLM) in Section 5.2. Finally we generalize our algorithm to nonlinear models in Section 6. To limit the memory in this general case we resort to the generalized Gauss-Newton approximation of the covariance matrix.

5.1 L-RVGA for linear regression

We first consider the simpler case of a linear model $y_t = x_t^T \theta + w_t$ where $w_t \sim \mathcal{N}(0, \sigma_w \mathbf{I})$. σ_w is a scaling factor we will set to 1, assuming outputs are normalized. The RVGA updates (35)-(36) are

equivalent to the following explicit updates in the linear case, see Lambert et al. 2021:

$$P_t^{-1} = P_{t-1}^{-1} + x_t x_t^T \tag{37}$$

$$\mu_t = \mu_{t-1} + P_t x_t (y_t - x_t^T \mu_{t-1}). \tag{38}$$

These updates are also known as recursive least squares or the recursive ridge regression. They match the updates of the linear Kalman filter with static state. Thus, the first stage approximation (10) is explicitly solved. Let us turn to the second stage approximation (11), that addresses limited memory requirements. Using a factorized limited approximation of the precision matrix, the latter equations become

$$P_t^{-1} = W_{t-1} W_{t-1}^T + \Psi_{t-1} + x_t x_t^T \underset{FA}{\approx} W_t W_t^T + \Psi_t \tag{39}$$

$$\mu_t = \mu_{t-1} + (W_t W_t^T + \Psi_t)^{-1} x_t (y_t - x_t^T \mu_{t-1}), \tag{40}$$

where we have used the symbol $\underset{FA}{\approx}$ to express the factor analysis approximation given by one outer step of the recursive EM Algorithm 1, whose computational and storage costs are linear in d. Regarding parameter μ, the update (40) can be rewritten using the Woodbury formula as follows:

$$\begin{aligned}\mu_t &= \mu_{t-1} + (\Psi_t^{-1} - \Psi_t^{-1} W_t M_t^{-1} W_t^T \Psi_t^{-1}) x_t (y_t - x_t^T \mu_{t-1}) \\ &= \mu_{t-1} + \Psi_t^{-1} (x_t - W_t M_t^{-1} (W_t^T \Psi_t^{-1} x_t))(y_t - x_t^T \mu_{t-1}),\end{aligned}$$

where $M_t = \mathbf{I}_p + W_t^T \Psi_t^{-1} W_t$. These operations are also linear in d with a memory requirement being linear in d. We have thus successfully implemented a limited-memory version of the two-stage RVGA (10)-(11) in the linear case.

5.2 L-RVGA for generalized linear models

Building upon our prior work on RVGA Lambert et al. (2021), the latter approach may be extended whenever the observation y_t follows an exponential family distribution, which may be advantageous to deal with classification problems. An exponential family distribution (Pitcher, 1979) takes the form:

$$p(y_t|\theta) = m(y_t) \exp(\eta^T y_t - F(\eta)), \tag{41}$$

where F is the log partition function and $m(y)$ is a normalization function. The natural parameter η is related to the expectation parameter $m = \mathbf{E}(y|\theta)$ with 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 θ through a nonlinear function $\eta = h(\theta, x)$. In the case of generalized linear models with a canonical natural parameter, this function is linear, that is, of the form $\eta = \theta^T x$. The RVGA updates (12)-(13) write in this case:

$$P_t^{-1} = P_{t-1}^{-1} + \gamma_t x_t x_t^T \tag{42}$$

$$\mu_t = \mu_{t-1} + P_{t-1} x_t \varepsilon_t, \tag{43}$$

where we have introduced the scalar weighting parameter γ_t and the scalar error ε_t which depends on θ as follows:

$$\gamma_t = \mathbf{E}_{\theta \sim \mathcal{N}(\mu_t, P_t)}[\mathrm{Cov}(y_t|\theta)] \tag{44}$$

$$\varepsilon_t = y_t - \mathbf{E}_{\theta \sim \mathcal{N}(\mu_t, P_t)}[g^{-1}(x_t^T \theta)], \tag{45}$$

and where we have used the relation $\nabla^2_\eta F(\eta(\theta)) = \mathrm{Cov}(y|\theta)$. The factorized approximation of this update can be derived similarly to the linear case, as we only need to use a weighted version of our factor analysis approximation $W_{t-1}W_{t-1}^T + \Psi_{t-1} + \gamma_t x_t x_t^T \underset{FA}{\approx} W_t W_t^T + \Psi_t$. The computation of the scalar terms γ_t and ε_t may require resorting to approximations. Those approximations may rely on the tools developed in the next subsection, devoted to the general case. However, in the case of logistic regression these parameters were shown to be easily obtainable numerically and will serve as a baseline for our experimentations in Section 7.3.

6 Limited memory RVGA (L-RVGA) : general case

In the general case the RVGA updates (12)-(13) do not have a closed form. Using a factorized limited approximation of the covariance at time step $t-1$, the update at step t then writes:

$$P_t^{-1} = W_{t-1}W_{t-1}^T + \Psi_{t-1} - \mathbf{E}_{\theta\sim\mathcal{N}(\mu_t,P_t)}[\nabla^2_\theta \log p(y_t|\theta, x_t)] \tag{46}$$

$$\mu_t = \mu_{t-1} + P_{t-1}\mathbf{E}_{\theta\sim\mathcal{N}(\mu_t,P_t)}[\nabla_\theta \log p(y_t|\theta, x_t)]. \tag{47}$$

Its implementation raises the following difficulties:

1. Computing μ_t and P_t^{-1} in (46)-(47) involves computing an expectation over a distribution parameterized by μ_t and P_t (the scheme is implicit).

2. As no closed form is generally available for the computation of the expectations $\mathbf{E}_{\theta\sim\mathcal{N}(\mu_t,P_t)}$, 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, $P_{t-1}^{-1} = W_{t-1}W_{t-1}^T + \Psi_{t-1}$ and we need to sample from a Gaussian using this structured precision matrix without storing (or inverting) a $d \times d$ matrix.

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

4. Finally, once the right-hand side of (46) is available, one must approximate it with a low-rank plus diagonal matrix $W_t W_t^T + \Psi_t$ using factor analysis. This point which corresponds to the second stage of our approximation procedure was thoroughly addressed in the previous section.

In the remainder of the section, we address the first three points.

6.1 Using extra-gradients for the implicit scheme

In this paragraph we address the Point 1 above and seek to make explicit the following scheme:

$$\mu_t = \mu_{t-1} - P_{t-1}\mathrm{E}_{\theta\sim\mathcal{N}(\mu_t,P_t)}[\nabla_\theta \ell_t(\theta)]$$
$$P_t^{-1} = P_{t-1}^{-1} + \mathrm{E}_{\theta\sim\mathcal{N}(\mu_t,P_t)}[\nabla^2_\theta \ell_t(\theta)],$$

which involves taking expectation of the gradient and the Hessian of the loss under the unknown Gaussian distribution we precisely seek to estimate. Although a closed form is available for linear and logistic regression, see Lambert et al. 2021, the general case requires resorting to an explicit

scheme. To this aim, we propose to update first the covariance and then the mean and to iterate the updates a second time:

Iterated RVGA

$$
\begin{aligned}
\hat{\mathbf{P}}_{\mathbf{t}}^{-1} &= P_{t-1}^{-1} - \mathrm{E}_{\theta\sim\mathcal{N}(\mu_{t-1},P_{t-1})}[\nabla_\theta^2 \log p(y_t|\theta)] \\
\hat{\mu}_{\mathbf{t}} &= \mu_{t-1} + \hat{\mathbf{P}}_{\mathbf{t}}\mathrm{E}_{\theta\sim\mathcal{N}(\mu_{t-1},P_{t-1})}[\nabla_\theta \log p(y_t|\theta)] \\
\mathbf{P}_{\mathbf{t}}^{-1} &= P_{t-1}^{-1} - \mathrm{E}_{\theta\sim\mathcal{N}(\hat{\mu}_{\mathbf{t}},\hat{\mathbf{P}}_{\mathbf{t}})}[\nabla_\theta^2 \log p(y_t|\theta)] \\
\mu_t &= \mu_{t-1} = \mathbf{P}_{\mathbf{t}}\mathrm{E}_{\theta\sim\mathcal{N}(\hat{\mu}_{\mathbf{t}},\hat{\mathbf{P}}_{\mathbf{t}})}[\nabla_\theta \log p(y_t|\theta)].
\end{aligned}
\tag{48}
$$

This scheme reminds of the extended iterated Kalman filter (Jazwinski, 1970), albeit much different. 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:

$$
f(\mu, P) = \mathbf{E}_{\mathcal{N}(\mu,P)}[\log p(y|\theta)]. \tag{49}
$$

The proof has been moved to Appendix F.1. Mirror Prox was initially proposed with a step size parameter and several works (Bach and Levy, 2019, Babanezhad and Lacoste-Julien, 2020) have proven convergence results for particular tuning of this parameter when the function f is convex. Our function is not convex in P but is convex in C where $P = CC^T$ is the Cholesky decomposition of P, this property has been shown by Challis and Barber (2013) and is recalled in Appendix F.2. Our setting does not use an explicit step size but uses instead a couple of two updates which allow for automatic adaptation to the local curvature. 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 Section 7.4.1).

6.2 Gaussian sampling from the precision matrix

In this paragraph we address Point 2 above. Suppose one wants to sample from a Gaussian $\mathcal{N}(0, (WW^T + \Psi)^{-1})$, where W is $d \times p$ matrix with $p \ll d$ and Ψ is diagonal, and the dimension d is too large to allow for any $d \times d$ matrix manipulation or storage. The method we propose is then as follows. We depart from the Woodbury inversion formula to retrieve the covariance

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

and seek to sample from the corresponding distribution $\mathcal{N}(0, P)$. The difficulty comes from the minus sign. A way around this problem is to note that the covariance matrix above may be interpreted as a conditioned Gaussian. Indeed we can use the following property (used already in the E-step for the EM algorithm for factor analysis):

Lemma 2. *Let us consider the following random variables $x \sim \mathcal{N}(0, \Psi^{-1})$, $\varepsilon \sim \mathcal{N}(0, \mathrm{I}_p)$ and z defined by $z = W^T x + \varepsilon$. Then we have from Gaussian conditioning formulas:*

$$
\begin{aligned}
P = (WW^T + \Psi)^{-1} &= \Psi^{-1} - \Psi^{-1}W(\mathrm{I}_p + W^T\Psi^{-1}W)^{-1}W^T\Psi^{-1} && (50)\\
&= \mathrm{Cov}(x) - \mathrm{Cov}(x,z)\mathrm{Cov}(z|x)^{-1}\mathrm{Cov}(z,x) && (51)\\
&= \mathrm{Cov}(x|z), && (52)
\end{aligned}
$$

which shows that P is the covariance of the conditional distribution $p(x \mid z = 0)$, or any other value for z as the conditional covariance does not depend on it.

We can use this lemma to estimate P through Monte-Carlo sampling on x and where importance weighting allows for conditioning. Indeed from the Bayesian formula:

$$p(x|z=0) \ \propto \ k(x)p(x),$$

where $k(x) = \mathcal{N}(0|W^T x, \mathrm{I}_p)$ and $p(x) = \mathcal{N}(x|0, \Psi^{-1})$. The detailed process to estimate P with importance sampling is inspired from the particle filter method (Gordon et al., 1993) and is derived as follows. First we sample K points $x_1, \ldots, x_K$ from $\mathcal{N}(0, \Psi^{-1})$. Then we compute the weights $w_i = k(x_i)$ for $1 \leq i \leq K$, and normalize the weights as the conditional distribution is known up to a constant, that is, we compute $w_i \leftarrow w_i/(\sum_1^K w_i)$. We thus derive an estimator of $\mathbf{E}[f(X)] \approx \sum_1^K w_i f(x_i)$ for any function f, in particular we have $P = \mathbf{E}[XX^T] \approx \sum_1^K w_i x_i x_i^T$. Mishkin et al. (2018) have also implemented fast Gaussian sampling using the factor analysis parameters on the precision matrix but their method requires computing two Cholesky decompositions on the latent space whereas our method is in $O(dK)$.

6.3 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 outer products. When d is large, the Hessian matrix that appears in the quantity we seek to approximate may be impossible to store. To apply the recursive EM, in a memory efficient way, we need to express this matrix as a sum of outer products. 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). If the probability belongs to an exponential family, we rather consider the Generalized Gauss-Newton (GGN) approximation (Martens, 2014) which exploits the structure of the loss as a composition of functions, and better approximates the local curvature (Kunstner et al., 2019).

We assume the conditional distribution to be of the form (41), where the natural parameter satisfies $\eta = h(\theta, x)$ with h a nonlinear function, like a neural network. To introduce the GGN approximation for our problem, we recall results from Martens 2014 and Ollivier 2018, that we have extended for our Bayesian framework:

Lemma 3. *Assume $p(y|\theta)$ is an exponential family (41) with a natural parameter η parametrized by θ such that $\eta = h(\theta)$ where h is any nonlinear function and y_t is a sample from this distribution. Consider the following approximation, where y_t is replaced by its expectation:*

$$-\mathbf{E}_\theta[\nabla_\theta^2 \log p(y_t|\theta)] \approx -\mathbf{E}_\theta \mathbf{E}_{y\sim p(y|\theta)}[\nabla_\theta^2 \log p(y|\theta)] := \mathbf{E}_\theta[F(\theta)], \tag{53}$$

where F denotes the Fisher information matrix. This approximation is equivalent to the (averaged) generalized Gauss-Newton approximation, which has the advantage of avoiding storage and use of $d \times d$ matrices, as the output y_t is generally of much smaller dimension than the parameter θ. We have indeed

$$-\mathbf{E}_\theta \mathbf{E}_{y\sim p(y|\theta)}[\nabla_\theta^2 \log p(y|\theta)] = \mathbf{E}_\theta[F(\theta)] = \mathbf{E}_\theta[\frac{\partial h}{\partial \theta} \times \mathrm{Cov}(y|\theta) \times \frac{\partial h}{\partial \theta}].$$

Moreover, the approximation (53) is exact if distribution $p(y|\theta)$ follows a generalized linear model (GLM).

The proof is recalled in Appendix D. Under the GGN approximation, the covariance update no longer depends on the labels y_t and only depends on the inputs data x_t as in the linear case. Assuming we can sample from the distribution of interest we denote by $\mathcal{N}(\mu, P)$, we may replace

y_t by its expectation and approximate the expectation through sampling, leading to the following outer product approximation:

$$P_t^{-1} \underset{\text{GGN}}{\approx} W_{t-1}W_{t-1}^T + \Psi_{t-1} + \mathbf{E}_{\theta\sim\mathcal{N}(\mu,P)}[\frac{\partial h}{\partial \theta} \times \text{Cov}(y|\theta) \times \frac{\partial h}{\partial \theta}] \tag{54}$$

$$\underset{\text{Sampling}}{\approx} W_{t-1}W_{t-1}^T + \Psi_{t-1} + \frac{1}{K}\sum_{i=1}^{K} \frac{\partial h}{\partial \theta}(\theta_i)\sqrt{\text{Cov}(y|\theta_i)^{-1}}\sqrt{\text{Cov}(y|\theta_i)^{-1}}\frac{\partial h}{\partial \theta}^T(\theta_i) \tag{55}$$

$$= W_{t-1}W_{t-1}^T + \Psi_{t-1} + \frac{1}{K}\sum_{i=1}^{K} c_i c_i^T \quad \text{where} \quad c_i = \frac{\partial h}{\partial \theta}(\theta_i)\sqrt{\text{Cov}(y|\theta_i)^{-1}} \tag{56}$$

$$= W_{t-1}W_{t-1}^T + \Psi_{t-1} + U_t U_t^T \quad \text{where} \quad U_t = \frac{1}{\sqrt{K}}\left(c_1 \cdots c_K\right) \tag{57}$$

$$\underset{\text{FA}}{\approx} W_t W_t^T + \Psi_t, \tag{58}$$

where the last line refers to the FA approximation developed in Section 4 applied to the mini-batch matrix U_t of size $d \times K$. As long as $K \ll d$ we see the memory cost is kept linear in d.

The full updates equations including the update for the mean (47) are detailed in Appendix E.

7 Experiments

Experiments have been performed on synthetic data for different classes of problems, from simpler linear problems to more difficult nonlinear problems. First of all, we focus on checking the improvements in memory requirements of our approach. The first class of problems addressed in Section 7.1 deals with the approximation of large scale covariance matrices too large to fit in memory. We show our approach requires far less memory than the batch EM or all classical online EM algorithms for factor analysis which require storing a $d \times d$ matrix in memory. These results directly apply to linear regression where we propose cheap updates for recursive linear regression in high dimensional spaces in Section 7.2. We consider then the logistic regression problem for which we can solve the recursive variational scheme in closed form at each step. This recursive variational scheme is compared to the Laplace approximation in Section 7.3. Our limited memory approach provides an estimation of the covariance without resorting to post-processing as in Laplace approximation. Finally we turn to more general cases in Section 7.4 where we propose an extra-gradient method to approximate the implicit scheme and importance sampling to approximate the expectations involved in the recursive variational scheme. The logistic regression problem, where we know how to compute the expectation analytically, offers a baseline for comparison. We evidence that the extra-gradient method combined with importance sampling allow for reaching our baseline results in terms of KL divergence. This is promising regarding the generalization of our algorithms to arbitrary nonlinear problems.

The experiments conducted may be summarized as follows:

- Application of the recursive EM to limited-memory online approximation of large scale covariance matrix via factor analysis (Section 7.1).
- Application of this large scale approximation to limited memory online linear regression (Section 7.2).
- Application of this large scale approximation to limited memory online logistic regression (Section 7.3).
- Generalization of this approach to any nonlinear models using extra-gradients and (importance) sampling (Section 7.4).

7.1 Application to the approximation of large-scale covariance matrix

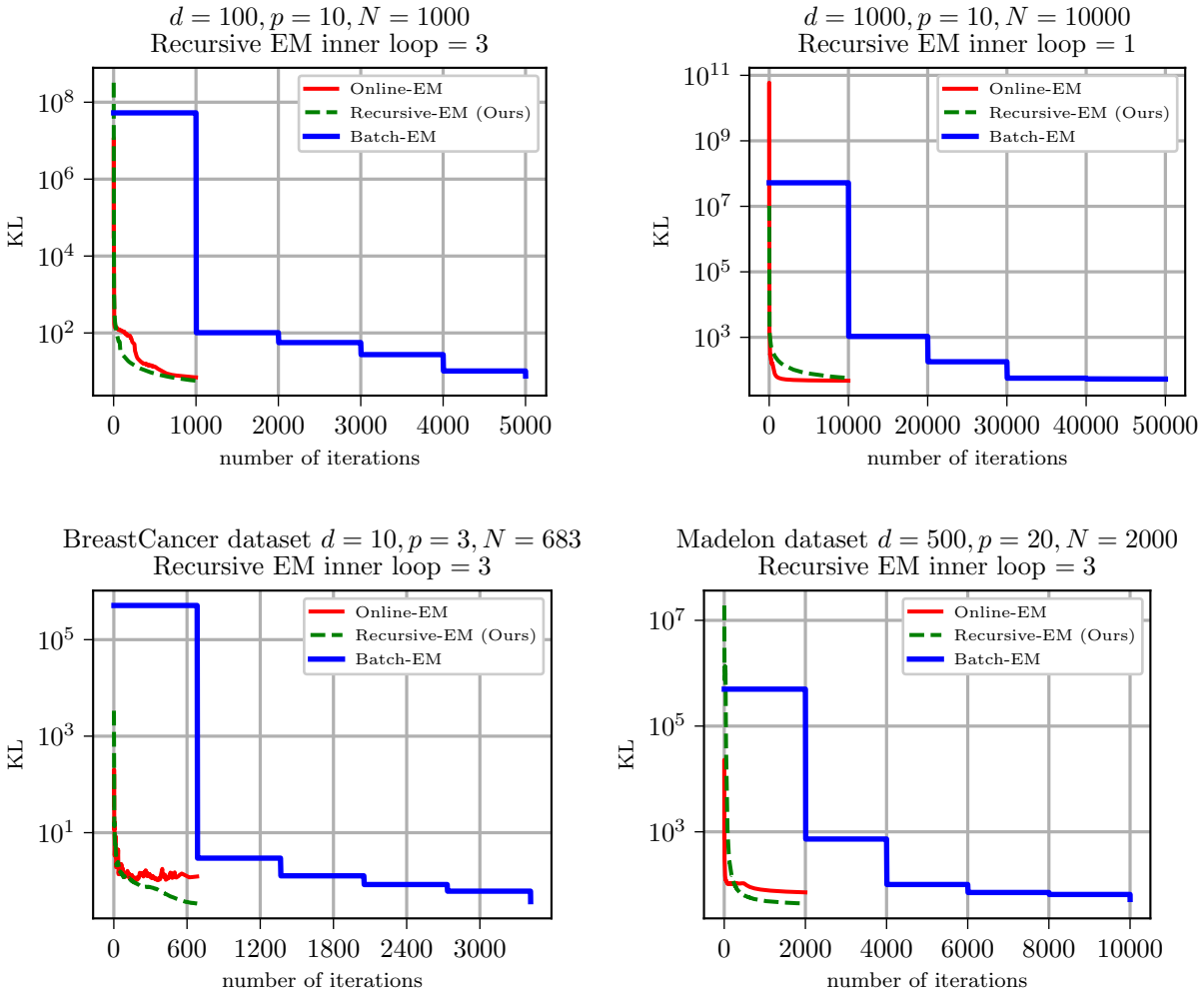

Figure 1: Factorization of a synthetic large scale covariance matrix of dimension $d = 100$ (upper left plot) and $d = 1000$ (upper right plot). The number of observations/samples is $N = 10d$ and we consider $p = 10$ for the dimension of the latent factors. The lower plots show additional results on 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 orange plain line the online EM tuned with a learning rate $\gamma = \frac{1}{t^{0.6}}$ (see Appendix A.4 for implementation details). Both of these algorithms have limited memory and converge after only one pass through the observations. We show also the batch EM, in solid blue line, after 5 passes through the observations. 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 can use only 1 inner loop for the recursive EM to reach a loss near the online EM. Moreover, the recursive EM gives better results than the other algorithms on real dataset as illustrated in lower left plot and lower right plot. The cost in time and memory of these algorithms are detailed in Table 1 for the higher dimensional case (upper right plot).

In this section, we assess the limited-memory recursive EM algorithm on large scale covariance

matrices. To approximate a covariance matrix we consider a slightly different version of Algorithm 1 developed in Section 4.2 for RVGA. Indeed the covariance updates in RVGA, see (4), are additive and the precision matrix P^{-1} keeps increasing. In covariance approximation, we seek to estimate P and not P^{-1}. This variant is detailed in Appendix A.3 and used in this section only.

To assess the method, we generate the data from an actual low-rank covariance plus diagonal matrix $S = \text{Diag}(\psi)+WW^T$ 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 ψ is a vector of size d. We then generate N samples from the zero-mean Gaussian distribution equipped with this covariance matrix S. The samples are normalized such that their norm is one on average. This normalization is an important part of the process and significantly helps the algorithms to converge. Another important setting is the initial values of the factors, since the data are normalized we have considered very low values $\varepsilon = \sigma_0^2 = 10^{-6}$ where ε and σ_0 are defined in algorithm 1. 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 A.4. While our method is parameter free it yields similar results to the online EM on our synthetic dataset as shown in upper plots of figure 1. Moreover we have found it yields much better results than the online EM as well as the batch EM on real datasets as shown in lower 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. For example for a dimension $d = 1000$, with a latent variable p in dimension 10, we save a factor 100 in memory and this reduction increases with the dimension. To assess that feature, we have monitored the processing and memory cost of our implementation. The results of the performance monitoring are shown in Table 1.

Algorithm	KL-divergence	Nb. iterations	Time per iteration	Memory cost
Recursive EM-1 inner loop	57	N	0.0012 s (for 1 inner loops)	**0.08 MB**
Online EM-1 inner loop	43	N	0.0005 s (for 1 inner loops)	**0.16 MB**
Batch EM	53	5	0.01 s	**8 MB**

Table 1: **Large scale matrix factorization with** $d = 1000$**,** $N = 10000$ **and** $p = 10$: performances associated to the right plot of figure 1. Recursive EM with one inner loop and online EM gives comparable results in terms of computation and memory cost whereas Batch EM needs high memory.
This test has been done with a CPU Intel core i7 at 2.3 GHz (without using GPU).

7.2 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 5.1. In the linear general case, we have an analytical form of the solution given by the linear Kalman filter which constitutes our baseline.

The input samples are generated using a Gaussian distribution of covariance C of size $d \times d$

encoded through a parameter c which drives the condition number as follows ($c = 1$ if not precised):

$$C = M^T \text{Diag}(1, 1/2^c, \cdots, 1/d^c) M, \tag{59}$$

where M is an orthogonal rotation matrix. Since the matrix C is rotated, the latent variables will not be directly available in a factor analysis form. The inputs are normalized in mean as for the previous section.

The outputs are generated with a normal noise:

$$y_t = x_t^T \theta^* + \varepsilon \text{ with } \varepsilon \sim \mathcal{N}(0, 1). \tag{60}$$

The outputs are also normalized such that we control the norm of the unknown parameter θ^*. This allows us to consider a strong prior information $P_0 = \sigma_0 \mathbf{I}_d$ with $\sigma_0 = 1$ for θ.

We first check that with $d = p$, where p is the dimension of the latent space in factor analysis, our algorithm converges well to the exact Kalman filter. It is the case indeed, see Figure 2, where we see that the recursive EM converges to the recursive Kalman optimal solution with a moderate number of inner loops (see the algorithm (1) for a precise definition of the inner loops). For all the following experiments, we will consider only 3 inner loops as it is often sufficient to obtain good convergence results even in high dimension.

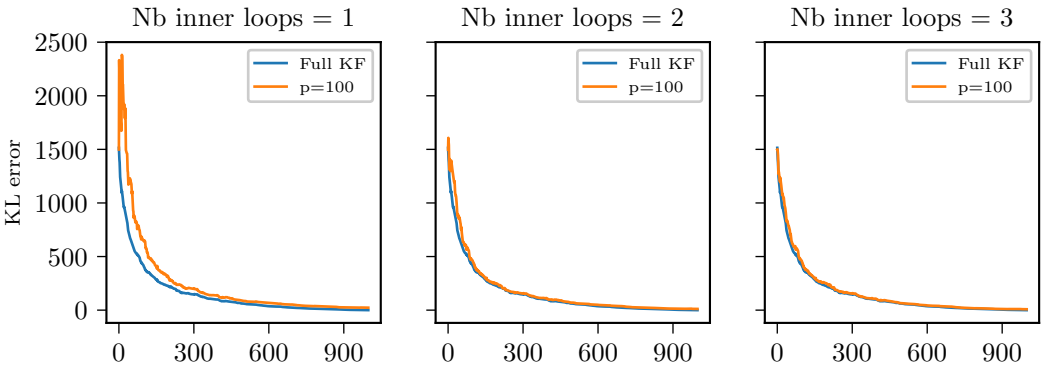

Figure 2: Sensitivity to the number of inner loops. Evolution of the KL between the posterior and its online approximation, for the exact linear Kalman (LKF), in blue, and the LRVGA (LSKF-EM), in orange. The L-RVGA matches the optimal Kalman filter when $d = p$ with a limited number of inner loops.

To assess how the approximation degrades the results, we have made experiments for different values of the latent space dimension p. We see in Figure 3 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. When reducing the dimension by a as high as a 200 factor, we still obtain a solution quite close to the optimal ones given by the full linear Kalman filter. These results, as well as the processing time and memory cost, are detailed in Table 2.

7.3 Application to Logistic regression

We now apply the covariance approximation to derive a limited memory version on a logistic regression problem. In the logistic case the covariance update is coupled with the mean update and we need to consider a weighted covariance matrix $S_N = \sum_{i=1}^N \gamma_i x_i x_i^T$ where γ_i depends on the previous

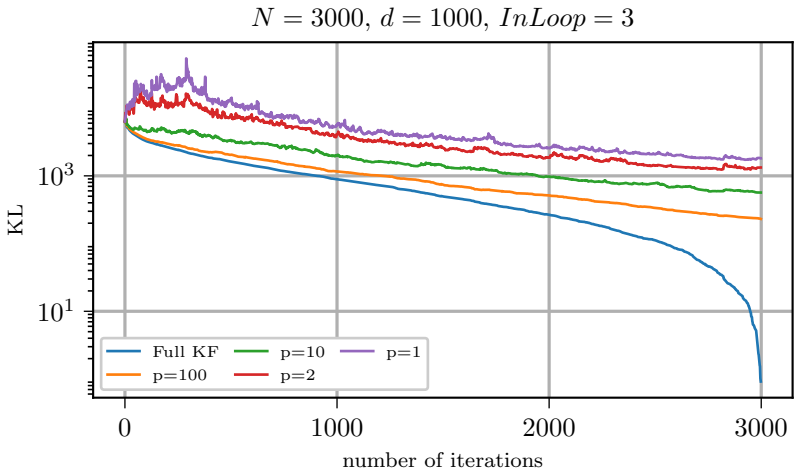

Figure 3: Sensitivity to the latent dimension p. The full Kalman error converges exactly to 0, the Kalman filter finding the posterior. The approximated version with $p = 100$ runs a close second. When p is very low, we observe some oscillations.

Algorithm	KL-divergence	Nb. iterations	Time per iteration	Memory cost
Full Kalman	0	1000	0.002 s	**8 MB**
L-RVGA ($p = 100$)	230	N	0.009 s (for 3 inner loops)	**0.8 MB**
L-RVGA ($p = 10$)	570	N	0.001 s (for 3 inner loops)	**0.09 MB**
L-RVGA ($p = 2$)	1340	N	0.0009 s (for 3 inner loops)	**0.03 MB**
L-RVGA ($p = 1$)	1837	N	0.0005 s (for 3 inner loops)	**0.02 MB**

Table 2: **Large scale linear regression with $d = 1000$, $N = 3000$ with 3 inner loops**: performance associated with the case illustrated on Figure 3. When p is low the cost in memory is drastically reduced with L-RVGA.

This test has been done with a CPU Intel core i7 at 2.3 GHz (without using GPU).

mean and covariance updates as described in Section 5.2. It turns out that for the logistic regression problem we can compute the γ_i's analytically (Lambert et al., 2021) using the approximation of the logistic function with the inverse probit function (Barber and Bishop, 1998a). The solution to the RVGA updates writes in this case:

$$k_t = \frac{\beta}{\sqrt{x_t^T P_t x_t + \beta^2}} \text{ where } \beta = \sqrt{\frac{8}{\pi}} \tag{61}$$

$$\mu_t = \mu_{t-1} + P_{t-1} x_t (y_t - \sigma(k_t x_t^T \mu_t)), \tag{62}$$

$$P_t^{-1} = P_{t-1}^{-1} + k_t \sigma'(k_t x_t^T \mu_t) x_t x_t^T, \tag{63}$$

where σ and σ' are the sigmoid function and its derivative. The unknown scalar parameters $\nu = x_t^T P_t x_t$ and $\alpha = x_t^T \mu_t$ can be found by solving a scalar fixed point equation with a Newton solver (see Lambert et al. 2021 for details).

We can apply our recursive EM algorithm on the input $u_t = x_t \sqrt{k_t \sigma'(k_t x_t^T \mu_t)}$ to do logistic regression in high dimension. To assess the performance, we generate N synthetic data using the

same method used in our previous work (Lambert et al., 2021). We consider a regularized problems where the prior is $P_0 = \sigma_0 \mathrm{I}_d$ with $\sigma_0 = 4$. This prior drives the smoothness of the loss function and the norm of the normal to the separator hyperplane. We do not have an exact filter to compare our results as in the linear case and our baseline is the unnormalized true posterior given by the empirical distribution attached to the data. The (unnormalized) divergence to this posterior is computed using the same sampling techniques than in our previous work (Lambert et al., 2021). Since the divergence is unnormalized it may take low values, which allows for relative comparison between algorithms but not absolute comparison.

Using this protocol, we compare the results for different values of the latent parameter, see Figure 4. 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 RVGA avoids them. Contrary to the linear case, even with $p = 1$, we obtain very good results. Table 3 shows the performance of the different algorithms, the L-RVGA algorithms have a much lower memory cost than the batch Laplace approximation.

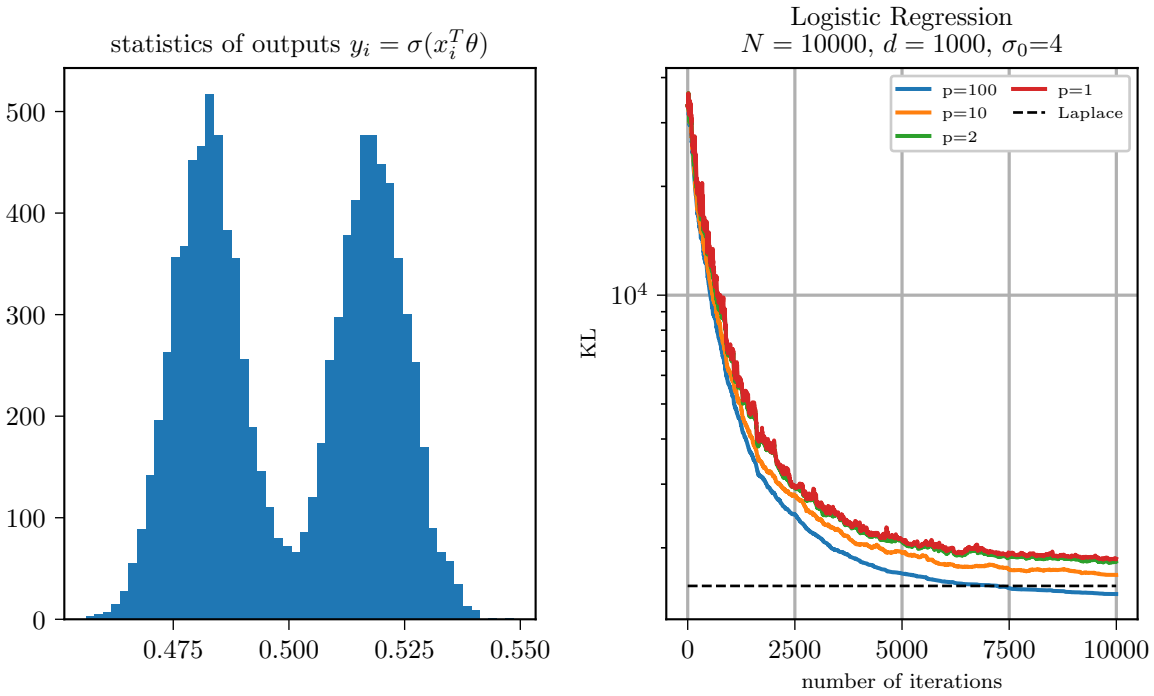

Figure 4: Sensitivity to the latent dimension p. The L-RVGA algorithms converge to lower divergence than the batch Laplace approximation even for $p = 1$. The estimated mean is roughly aligned with the maximum a-posteriori, additional observations may help to converge to a complete alignment. Only one inner loop is sufficient to ensure convergence in logistic regression.

7.4 Generalization of our approach

We address in this section the general nonlinear RVGA, based on the approximations detailed in Section 6. Since in the case of the logistic regression we have a closed form solution for RVGA, we may compare it to the approximated one proposed in the general case (by purposefully ignoring the closed form solution for comparison purposes). As the generalized Gauss-Newton approximation,

Algorithm	KL-divergence	Nb. iterations	TOTAL Time	Memory cost
Laplace	1568	13	16 s	**16 MB**
L-RVGA ($p = 100$)	1489	N	6 s (total time)	**0.8 MB**
L-RVGA ($p = 10$)	1681	N	5 s (total time)	**0.09 MB**
L-RVGA ($p = 2$)	1834	N	5 s (total time)	**0.03 MB**
L-RVGA ($p = 1$)	1864	N	5 s (total time)	**0.02 MB**

Table 3: **Large scale logistic regression with $d = 1000$, $N = 10000$ with 1 inner loop**: performance associated to the case illustrated on Figure 4. Only one inner loop used. These tests have been performed with a CPU Intel core i7 at 2.3 GHz (without using GPU).

presented in Section (6.3), is always exact in the logistic case, only the three others approximations will be considered. We first evaluate the extra-gradient approximation making the others approximations exact. We then assess the performance of the extra-gradient approximation combined with sampling and FA approximation.

7.4.1 Extra-gradients

In this paragraph we consider the approximation of the implicit scheme with an explicit scheme using extra-gradient, or Mirror prox, introduced in Section 6.1. To asses it we will make the other approximations exact, i.e., we use a full covariance matrix and an analytical computation of the expectation terms as described in Section 7.3. The expectations may be considered as exact even if they rely also on the approximation of the logistic function with the inverse probit function, as the difference proves negligible (Lambert et al., 2021). The extra-gradient scheme with analytical averaging reduces significantly the bias observed with the explicit scheme (without extra-gradient) and even with the implicit scheme. We illustrate this in a simple 2D case in Figure 5 where the performance of the extra-gradient scheme is compared to the RVGA implicit and explicit schemes as well as the natural gradient (Amari, 1996), and proves better in terms of discrepancy with the maximum a posteriori confidence ellipsoid. Results in higher dimension are shown in Appendix F.3.

7.4.2 Extra-gradients + factor analysis

When the covariance is approximated, the extra covariance update can make the Mirror prox scheme unstable. We have observed it is preferable to skip the extra covariance update when using approximation of the covariance with factor analysis, see Appendix F.3.

7.4.3 Extra-gradients + factor analysis + importance sampling

The last approximation we may consider is sampling where we replace the analytical average with sampled averaged. This sampled average will give less precise results in the logistic regression case but can be used in any class of problems and allows for generalization of our algorithm. In the case of RVGA with logistic regression we need to compute the expectation of the gradient and the Hessian of the logistic loss:

$$\mathbf{E}_\theta[\nabla_\theta \ell_t(\theta)] = -y_t x_t + \mathbf{E}_\theta[\sigma(x_t^T\theta)]x_t \tag{64}$$

$$\mathbf{E}_\theta[\nabla^2_\theta \ell_t(\theta)] = \mathbf{E}_\theta[\sigma(x_t^T\theta)(1-\sigma(x_t^T\theta))]x_t x_t^T. \tag{65}$$

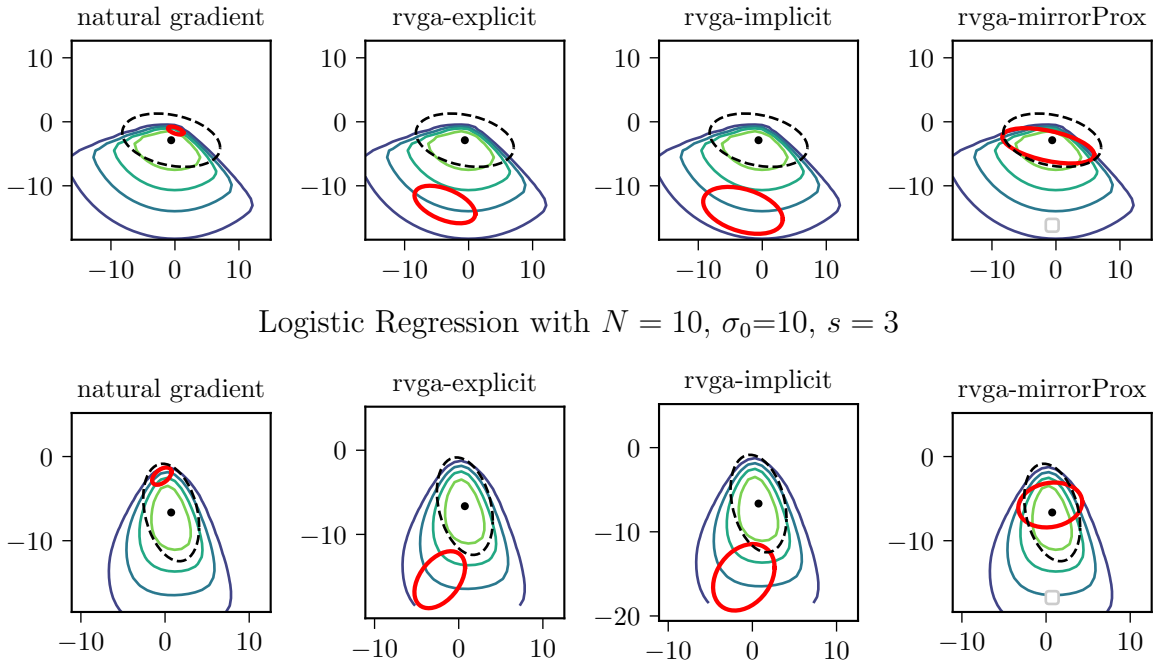

Figure 5: Data with a sharp prior $\sigma_0 = 10$. Confidence ellipsoids at final time for various algorithms (in red). The contour lines of the true posterior, which is known up to a constant, are displayed. The batch Laplace ellipsoid is shown in dashed line. The extra-gradient schemes have a divergence comparable to the implicit scheme but look closer to the maximum a posteriori.

Ignoring that we know in this case how to find a closed form solution, the integral terms are then approximated with sampling for comparison purposes:

$$\mathbf{E}_\theta[\sigma(x_t^T\theta)] \approx \sum_{i=1}^{K} w_i \sigma(x_t^T\theta_i) \tag{66}$$

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

where the samples θ_i are drawn from the Gaussian distribution $\mathcal{N}(\mu, (\Psi + WW^T)^{-1})$ in an efficient way using importance sampling (the w_i's denote the weights) as described in Section 6.2.

Approximating the average with sampling may degrade the convergence. However if we combine sampling with the extra-gradient version previously described we can restore the convergence as shown in Figure 6. Moreover, we have found few samples may be sufficient to provide good convergence of the algorithm in term of KL as shown in Figure 7.

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

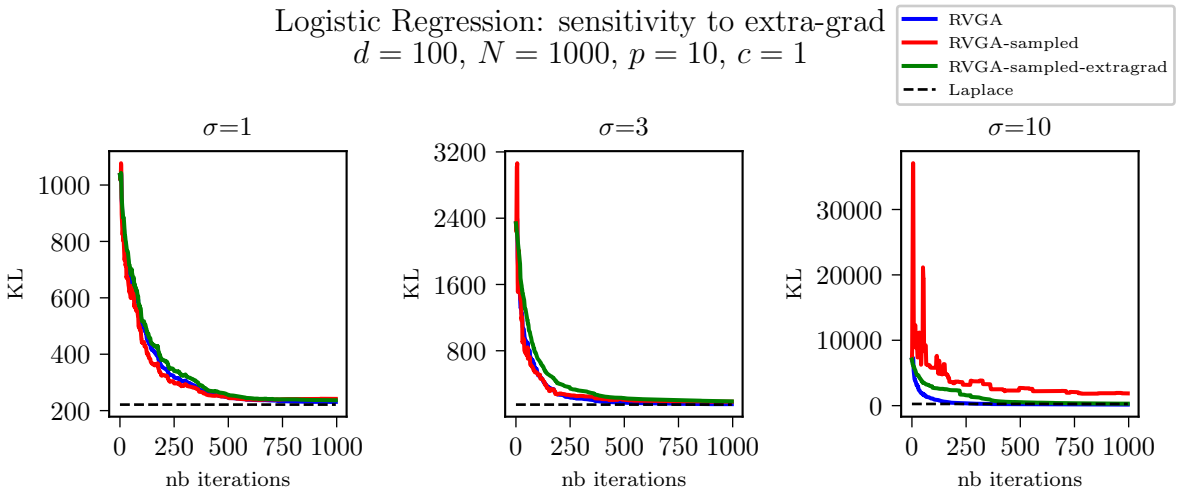

Figure 6: KL divergence to the true posterior. Note that in all graphics related to logistic regression, the posterior is known up to a constant, so that the KL are known up to a constant. As a result only the relative KLs between the algorithms are meaningful, not their absolute values. The graphics displays comparisons of the implicit RVGA where the expectations are computed analytically for the logistic regression and the explicit RVGA where the expectations are approximated with sampling ($K = 10$ samples are used). Two versions of the explicit scheme are considered: the first version does not use extra-gradients, the second one uses a Mirror prox (where we have skipped the extra covariance update). We consider the same factor analysis approximation $p = 10$ for all the algorithms with different values of prior: from smooth prior ($\sigma = 1$) to sharp prior ($\sigma = 10$). The smoothness of the prior drives the smoothness of the posterior distribution. All the algorithms behave equally when the prior is smooth. When the prior becomes sharper, extra-gradients help to stabilize the scheme.

Conclusion

We have developed a new second order algorithm, called L-RVGA, for online variational inference which scales to both large data set and high dimension. This algorithm is built on a two stage variational problem which combines a variational Gaussian approximation followed by a factor analysis approximation of the covariance (precision) matrix. L-RVGA is able to estimate the mean and the covariance of the distribution of the latent parameters in a memory efficient and paremeter-free way and with only one pass through the data. We have tested it on a linear and logistic regression problem and shown how to extend it to more general nonlinear problems with extra-gradients, memory efficient sampling and the generalized Gauss Newton approximation. The L-RVGA has a much less cost in time and memory than other second order algorithms like the Kalman filter, the Laplace approximation or the natural gradient while staying stable even in the presence of strong nonlinearities such as logistic regression with distants clusters.

To build our generic algorithm we have introduced two new tools: a recursive EM algorithm for factor analysis, derived from a fixed point equation, 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. Finally, we have highlighted in experiments that the combination of sampling and extra-gradient increases significantly the stability of our generic algorithm. While our target distribution is unimodal, the L-RVGA algorithm may provide a building block for the approximation

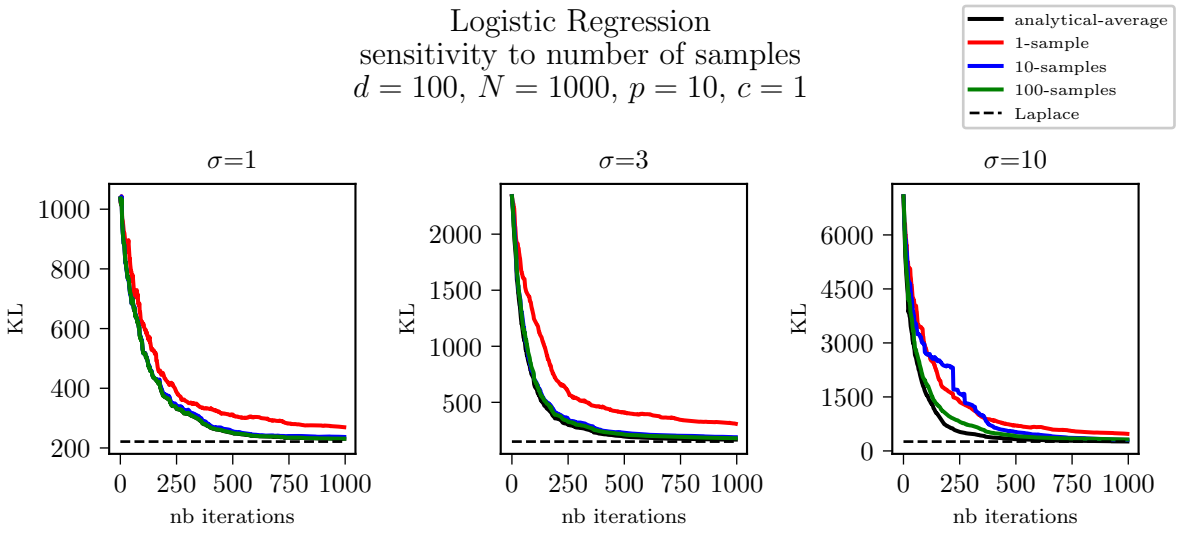

Figure 7: Comparison of the explicit RVGA where the expectations are computed analytically (for the logistic regression) and the explicit RVGA where the expectations are approximated with sampling for 1, 10 and 100 samples. All the algorithms use a Mirror prox (where we have skipped the extra covariance update) and use the same factor analysis approximation $p = 10$. 10 samples are sufficient even in the sharper case to converge to the batch Laplace KL.

of more generic posterior distributions in high dimension. Moreover it may be used in the context of stochastic optimization keeping only the mean and considering the prior as a regularizer. This version might compete with 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

Shun-Ichi Amari. Neural learning in structured parameter spaces: Natural riemannian gradient. *International Conference on Neural Information Processing Systems*, pages 127–133, 1996.

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

Haran Arasaratnam and Simon Haykin. Cubature Kalman filters. *IEEE Transactions on Automatic Control*, 54:1254 – 1269, 2009.

Reza Babanezhad and Simon Lacoste-Julien. Geometry-aware universal mirror-prox. *arXiv:2011.11203*, 2020.

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.

Axel Barrau and Silvere Bonnabel. Navigating with highly precise odometry and noisy GPS: a case study. *IFAC-PapersOnLine*, 49(18):618–623, 2016.

Axel Barrau and Silvere Bonnabel. Invariant Kalman filtering. *Annual Review of Control, Robotics, and Autonomous Systems*, 1:237–257, 2018.

Stephen Becker, Jalal Fadili, and Peter Ochs. On quasi-newton forward-backward splitting: Proximal calculus and convergence. *SIAM J. Optim.*, 29(4):2445–2481, 2019.

R. H. Byrd, S. L. Hansen, J. Nocedal, and Y. Singer. A stochastic quasi-newton method for large-scale optimization. *arXiv:1401.7020*, 2015.

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.

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.

N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proc. F Radar Signal Process. UK*, 140(2):107, 1993.

Mohinder S. Grewal and Angus P. Andrews. Applications of Kalman filtering in aerospace 1960 to the present [historical perspectives]. *IEEE Control Systems Magazine*, 30(3):69–78, 2010.

H.H. Harman. *Modern Factor Analysis*. University of Chicago Press, 1967.

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.

Yann Lecun, Patrice Y. Simard, and Barak Pearlmutter. Automatic learning rate maximization by on-line estimation of the hessian's eigenvectors. *Advances in Neural Information Processing Systems*, 1993.

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.

Dong C. Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1):503–528, 1989.

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

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.

Frank Nielsen. An elementary introduction to information geometry. *Entropy*, 22(10), 2020.

E. Oja. A simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 15(3):267–273, 1982.

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.

G. V. Puskorius and L. A. Feldkamp. Decoupled extended Kalman filter training of feedforward layered networks. *International Joint Conference on Neural Networks*, 1991.

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.

Sebastian Thrun, Yufeng Liu, Daphne Koller, Andrew Y. Ng, Zoubin Ghahramani, and Hugh Durrant-Whyte. Simultaneous localization and mapping with sparse extended information filters. *The International Journal of Robotics Research*, 23(7-8), 2004.

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

Richard Von Mises and Hilda Pollaczek-Geiringer. Praktische verfahren der gleichungsauflösung . *Zamm-zeitschrift Fur Angewandte Mathematik Und Mechanik*, 9:152–164, 1929.

A The recursive EM

A.1 Recall on classical EM factor analysis

Given a Gaussian variable u assumed of zero mean, we want to approximate the empirical covariance matrix $\frac{1}{N}\sum_{i=1}^{N} u_i u_i^T = S_N$ with a "diagonal + low rank" structure $\Psi + WW^T$. We introduce a latent variable z such that:

$$u = Wz + \varepsilon \quad \text{with } \varepsilon \sim \mathcal{N}(0, \Psi) \quad \text{and } z \sim \mathcal{N}(0, \mathbf{I}_p). \tag{67}$$

As a consequence we have:

$$u|z \sim \mathcal{N}(Wz, \Psi) \tag{68}$$

$$u \sim \mathcal{N}(0, WW^T + \Psi), \tag{69}$$

and the covariance of u is the quantity we want to factorize. The probability of the latent variable z given u can be computed with the Bayes formulas:

$$z|u \sim \mathcal{N}(M^{-1}W^T\Psi^{-1}u, M^{-1}) \text{ where } M = \mathbf{I}_p + W^T\Psi^{-1}W, \tag{70}$$

which gives the E-step of EM computed for the current values of the matrix W and Ψ:

$$\mathbf{E}[z|u] = M^{-1}W^T\Psi^{-1}u \tag{71}$$

$$\mathbf{E}[zz^T|u] = M^{-1} + \mathbf{E}[z|u]\mathbf{E}[z|u]^T \tag{72}$$

The M-step of EM is given by the maximization of the total likelihood L with respect to W and Ψ:

$$L(W,\Psi) = \log p(u_1, z_1, \ldots, u_N, z_N|W,\Psi) = \log p(u_1, \ldots, u_N|z_1, \ldots, z_N, W, \Psi) + \log p(z_1, \ldots, z_N) \tag{73}$$

$$= -\frac{N}{2}\log\det(\Psi) - \frac{1}{2}\sum_{t=1}^{N}(u_t - Wz_t)^T\Psi^{-1}(u_t - Wz_t) + K, \tag{74}$$

where K is a constant that does not depend on W or Ψ,

$$= -\frac{N}{2}\log\det(\Psi) - \frac{1}{2}\sum_{t=1}^{N}Tr((u_tu_t^T + Wz_tz_t^TW^T - 2Wz_tu_t^T)\Psi^{-1}) + K \tag{75}$$

$$= -\frac{N}{2}\log\det(\Psi) - \frac{N}{2}Tr(S_N\Psi^{-1}) - \frac{1}{2}\sum_{t=1}^{N}Tr(W^T\Psi^{-1}Wz_tz_t^T) - \sum_{t=1}^{N}Tr(z_tu_t^T\Psi^{-1}W) + K \tag{76}$$

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

$$\frac{\partial}{\partial W}\mathbf{E}_{p(z|u)}[L(W,\Psi)] = -\sum_{t=1}^{N}\Psi^{-1}W\mathbf{E}[z_tz_t^T|u_t] - \sum_{t=1}^{N}\Psi^{-1}u_t\mathbf{E}[z_t|u_t]^T = 0 \tag{77}$$

$$\implies W = \sum_{t=1}^{N}u_t\mathbf{E}[z_t|u_t]^T(\sum_{t=1}^{N}\mathbf{E}[z_tz_t^T|u_t])^{-1}. \tag{78}$$

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

$$\Psi = \mathrm{diag}(\mathrm{S_N} + \frac{1}{\mathrm{N}}\sum_{\mathrm{t=1}}^{\mathrm{N}}\mathrm{W}\mathbf{E}[\mathrm{z_tz_t^T}|\mathrm{u_t}]\mathrm{W^T} - \frac{2}{\mathrm{N}}\sum_{\mathrm{t=1}}^{\mathrm{N}}\mathrm{W}\mathbf{E}[\mathrm{z_t}|\mathrm{u_t}]\mathrm{u_t^T}) \tag{79}$$

$$= \mathrm{diag}(\mathrm{S_N} - \mathrm{W_{new}}\frac{1}{\mathrm{N}}\sum_{\mathrm{t=1}}^{\mathrm{N}}\mathbf{E}[\mathrm{z_t}|\mathrm{u_t}]\mathrm{u_t^T}). \tag{80}$$

And finally the EM updates gives:

$$\textbf{E-Step:} \tag{81}$$

$$\mathbf{E}[z_t|u_t] = M^{-1}B^T\Psi^{-1}u_t$$

$$\mathbf{E}[z_t z_t^T|u_t] = M^{-1} + \mathbf{E}[z_t|u_t]\mathbf{E}[z_t|u_t]^T. \tag{82}$$

$$\textbf{M-Step:} \tag{83}$$

$$B_{new} = \sum_{t=1}^{N} u_t \mathbf{E}[z_t|u_t]^T (\sum_{t=1}^{N} \mathbf{E}[z_t z_t^T|u_t])^{-1}$$

$$\Psi_{new} = \mathrm{diag}(\mathrm{S_N} - \mathrm{W_{new}} \frac{1}{\mathrm{N}} \sum_{\mathrm{t=1}}^{\mathrm{N}} \mathbf{E}[\mathrm{z_t}|\mathrm{u_t}]\mathrm{u_t^T}). \tag{84}$$

A.2 Factor analysis with EM as a fixed point : proof of lemma 1

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:

Proof. The EM updates (31) and (32) can be rewritten in batch form where

$$X = \left(u_1 \cdots u_m\right),$$

is the sample observation matrix of size $m \times d$ such that $S_N = \frac{1}{N} X X^T$ and

$$Z = \left(z_1 \cdots z_m\right),$$

is the sample latent matrix of size $p \times m$:

E-Step:

$$\mathbf{E}[Z|X] = M^{-1}B^T\Psi^{-1}X \tag{85}$$

$$Cov[Z|X] = M^{-1} + \frac{1}{N}\mathbf{E}[Z|X]\mathbf{E}[Z|X]^T = M^{-1} + M^{-1}B^T\Psi^{-1}S_N\Psi^{-1}BM^{-1}. \tag{86}$$

M-Step:

$$B_{new} = \frac{1}{N} X \mathbf{E}[Z|X]^T (Cov[Z|X])^{-1} \tag{87}$$

$$\Psi_{new} = \mathrm{diag}(\frac{1}{\mathrm{N}}\mathrm{XX^T} - \mathrm{B_{new}}\frac{1}{\mathrm{N}}\mathbf{E}[\mathrm{Z|X}]\mathrm{X^T}). \tag{88}$$

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

$$\begin{aligned}
W_{new} &= \frac{1}{N} X \mathbf{E}[Z|X]^T (Cov[Z|X])^{-1} \\
&= S_N\Psi^{-1}WM^{-1}(M^{-1} + M^{-1}W^T\Psi^{-1}S_N\Psi^{-1}WM^{-1})^{-1} \\
&= S_N\Psi^{-1}W(\mathbf{I}_p + M^{-1}W^T\Psi^{-1}S_N\Psi^{-1}W)^{-1} \\
\Psi_{new} &= \mathrm{diag}(\frac{1}{\mathrm{N}}\mathrm{XX^T} - \mathrm{W_{new}}\frac{1}{\mathrm{N}}\mathbf{E}[\mathrm{Z|X}]\mathrm{X^T}) \\
&= \mathrm{diag}(\mathrm{S_N} - \mathrm{W_{new}M^{-1}W^T\Psi^{-1}S_N}) \\
&= \mathrm{diag}(\mathrm{S_N} - \mathrm{S_N\Psi^{-1}W(M + W^T\Psi^{-1}S_N\Psi^{-1}W)^{-1}W^T\Psi^{-1}S_N}),
\end{aligned}$$

which proves the lemma 1. □

A.3 The recursive EM: rationale for algorithm 1

If we replace S_N by the partial covariance matrix up to input u_t : $S_t = \frac{1}{t}\sum_{i=1}^{t} u_i u_i^T$ in the updates (33), we obtain an online version of EM which converges in only one pass through the data but which requires to store a $d \times d$ matrix with a number of operation in $O(d^2)$. Our limited memory recursive EM variant consists in replacing S_t by the online average $\frac{1}{t}u_t u_t^T + \frac{t-1}{t}(W_{t-1}W_{t-1}^T + \Psi_{t-1})$ at each time step t. This allows for a linear in d cost and memory requirement. Let:

$$\begin{aligned} V_t &:= S_t \Psi_t^{-1} W_t \\ &= (\frac{1}{t}u_t u_t^T + \frac{t-1}{t}W_{t-1}W_{t-1}^T + \frac{t-1}{t}\Psi_{t-1})\Psi_t^{-1}W_t \\ &= \frac{1}{t}u_t(u_t^T\Psi_t^{-1}W_t) + \frac{t-1}{t}W_{t-1}(W_{t-1}^T\Psi_t^{-1}W_t) + \frac{t-1}{t}\Psi_{t-1}\Psi_t^{-1}W_t, \end{aligned}$$

the update equations then become:

$$\begin{aligned} W_{new} &= S_t\Psi_t^{-1}W_t(\mathbf{I}_p + M_t^{-1}W_t^T\Psi_t^{-1}S_t\Psi_t^{-1}W_t)^{-1} \\ &= V_t(\mathbf{I}_p + M_t^{-1}W_t^T\Psi_t^{-1}V_t)^{-1} \\ \Psi_{new} &= \mathrm{diag}(S_t - W_{new}M_t^{-1}W_t^T\Psi_t^{-1}S_t) \\ &= \mathrm{diag}(\frac{1}{t}u_t u_t^T + \frac{t-1}{t}W_{t-1}W_{t-1}^T + \frac{t-1}{t}\Psi_{t-1}) - \mathrm{diag}(W_{new}M_t^{-1}V_t^T) \\ &= \frac{1}{t}u_t * u_t + \frac{t-1}{t}\sum_{i=1}^{p} W_{t-1} * W_{t-1}[.,i] + \frac{t-1}{t}\Psi_{t-1} - \sum_{i=1}^{p} W_{new}M_t^{-1} * V_t[.,i], \end{aligned}$$

where $x * y$ is a memory cheap component wise operation of two matrices x and y of same size $d \times p$. If $p = 1$ we have $x * y = \mathrm{diag}(xy^T)$, if $p > 1$ we have $\sum_{i=1}^{p} x * y[.,i] = \mathrm{diag}(xy^T)$. For the first input at $t = 1$ we may consider W_0 and Ψ_0 as null matrix and W_1 and Ψ_1 as random with very small entries (no prior). For the following inputs at $t > 1$ we initialize the parameters with the previous estimate $W_t = W_{t-1}$ and $\Psi_t = \Psi_{t-1}$. Current and previous parameters appear in the updates and may not be confused: W_{t-1} and Ψ_{t-1} are remaining fixed through the inner iterations whereas W_t and Ψ_t are variable through the inner iterations.

In the recursive variational problem we have not a moving average update but an information update which keeps increasing. In this case we need to replace S_t by $u_t u_t^T + W_{t-1}W_{t-1}^T + \Psi_{t-1}$ without the averaging. This is this version which is used in RVGA and described in algorithm 1.

A.4 Derivation of the online EM algorithm for factor analysis

In this section we derive the online EM for the factor analysis problem to compare it to our Recursive EM algorithm. We use the same notation than in the previous section where $u_{i=1,\ldots,N}$ are our N observations in dimension d and $z_{i=1,\ldots,N}$ are our latent variables in dimension p. Moreover, we suppose that each couple of variables (z_t, u_t) are independent and belong to an exponential family given by $\log p(z_t, u_t|\theta) =< S(z_t, u_t), \phi(\theta) > -F(\theta)$, where F is the log partition function, ϕ is a function which map the natural parameter and $S(z_t, u_t)$ are the sufficient statistics.

Dempster et al. (1977) have shown in their seminal work that EM asymptotically converge when $N \to \infty$ to a (not necessary unique) stationary point θ^* of the following fixed point equation :

$$\theta = \arg\max_{\theta \in \Theta} \quad \mathbf{E}_{u_t}\mathbf{E}_{z_t \sim p(z|u_t,\theta)}[S(z_t, u_t)] = K(\theta). \tag{89}$$

This fixed point equation involve maximum of expectations not easily tractable in the general case. However, in the particular case of factor analysis, we have seen we can computed the fixed point in a closed form and build upon it a recursive EM algorithm. To avoid intractable expectation in the general case, the fixed point equation in θ can be rewritten as a fixed point equation with respect to the sufficient statistics S:

$$S = \mathbf{E}_{u_t}\mathbf{E}_{z_t\sim p(z|u_t,\theta^*(S))}[S(z_t,u_t)] = T(S) \tag{90}$$

$$\text{where } \theta^*(S) = \arg\max_{\theta\in\Theta} \sum_{t=1}^{N} < S(z_t,u_t), \phi(\theta) > -F(\theta). \tag{91}$$

The online EM algorithm (Cappé and Moulines, 2009) 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 θ are update online with an adaptive step γ_t at each new incoming observations u_t as follows:

$$S_t = (1-\gamma_t)S_{t-1} + \gamma_t \mathbf{E}_{z_t\sim p(z|u_t,\theta_{t-1})}[S(z_t,u_t)] \quad \text{(E-step)} \tag{92}$$

$$\theta_t = \arg\max_{\theta\in\Theta} < S_t, \phi(\theta) > -F(\theta) \quad \text{(M-step)}. \tag{93}$$

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

$$\log p(u_t, z_t) = -\frac{1}{2}\mathrm{Tr}[\Sigma^{-1}S(u_t,z_t)] - \frac{1}{2}\log\det\Sigma + c, \tag{94}$$

where the joint covariance is :

$$\Sigma = \begin{pmatrix} WW^T + \Psi & W \\ W^T & I_p \end{pmatrix}, \tag{95}$$

and the sufficient statistics are :

$$S(u_t,z_t) = \begin{pmatrix} u_t \\ z_t \end{pmatrix}\begin{pmatrix} u_t \\ z_t \end{pmatrix}^T = \begin{pmatrix} u_t u_t^T & u_t z_t^T \\ z_t u_t^T & z_t z_t^T \end{pmatrix}. \tag{96}$$

The expectation of the sufficient statistics gives

$$\mathbf{E}_{z_t\sim p(z|u_t,\theta_{t-1})}[S(u_t,z_t)] = \begin{pmatrix} u_t u_t^T & u_t\mathbf{E}[z_t^T|u_t] \\ \mathbf{E}[z_t|u_t]u_t^T & \mathbf{E}[z_t z_t^T|u_t] \end{pmatrix} \tag{97}$$

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

Finally, using the same formula than the batch EM in Appendix A.1, the online EM updates for factor analysis become:

Online EM

E-step

$$\begin{aligned}
S_{1.t} &= (1-\gamma_t)S_{1.t-1} + \gamma_t u_t u_t^T \\
S_{2.t} &= (1-\gamma_t)S_{2.t-1} + \gamma_t \mathbf{E}[z_t|u_t]u_t^T \\
&= (1-\gamma_t)\mu_{t-1} + \gamma_t M_{t-1}^{-1}W_{t-1}^T\Psi_{t-1}^{-1}u_t u_t^T \\
S_{3.t} &= (1-\gamma_t)S_{3.t-1} + \gamma_t \mathbf{E}[z_t z_t^T|u_t] \\
&= (1-\gamma_t)S_{3.t-1} + \gamma_t(M_{t-1}^{-1} + M_{t-1}^{-1}W_{t-1}^T\Psi_{t-1}^{-1}u_t u_t^T\Psi_{t-1}^{-1}W_{t-1}M_{t-1}^{-1})
\end{aligned} \tag{98}$$

M-step

$$\begin{aligned}
W_t &= S_{2.t}^T S_{3.t}^{-1} \\
\Psi_t &= \mathrm{diag}(\mathrm{S_{1.t}}) - \mathrm{diag}(\mathrm{W_t S_{2.t}}).
\end{aligned}$$

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:

$$\mathrm{diag(S_{1.t})} = (1-\gamma_t)\mathrm{diag(S_{1.t-1})} + \gamma_t \mathrm{u_t} * \mathrm{u_t}, \tag{99}$$

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

To choose the step size γ_t we must satisfy the Robins Monro rules:

$$\sum_{t=1}^{N} \gamma_t = \infty \quad \sum_{t=1}^{N} \gamma_t^2 < \infty. \tag{100}$$

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

$$\gamma_0 = 1 \tag{101}$$

$$\gamma_t = \frac{1}{t^{0.6}} \quad \forall t > 0. \tag{102}$$

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

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

$$\bar{W}_t = \frac{\tilde{t}-1}{\tilde{t}} W_{t-1} + \frac{1}{\tilde{t}} W_t \tag{103}$$

$$\bar{\Psi}_t = \frac{\tilde{t}-1}{\tilde{t}} \Psi_{t-1} + \frac{1}{\tilde{t}} \Psi_t. \tag{104}$$

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:

$$\max_{W,\Psi} \quad \log p(v_1, \ldots, v_N | W, \Psi) \text{ where } S_N = \frac{1}{N}\sum_{i=1}^{N} v_i v_i^T = P^{-1} \quad \text{(MLE)} \ . \tag{105}$$

$$\leq \max_{W,\Psi} \quad \mathrm{E}_z[\log p(v_1, \ldots, v_N, z | W, \Psi)] \text{ with } p(v \mid z) \sim \mathcal{N}(Wz, \Psi),\ p(z) \sim \mathcal{N}(0, I_p) \quad \text{(EM)} \ . \tag{106}$$

The advantage of the EM approach is that the total likelihood appearing in (106) 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 (105) 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 minimizing the Stein divergence D_S (dilated here by $\frac{1}{2}$ to be consistent with the likelihood loss):

$$\min_{W,\Psi} \quad \frac{1}{2} D_S(S, WW^T + \Psi) = \min_{W,\Psi} \quad \frac{1}{2}\mathrm{Tr}(S(WW^T + \Psi)^{-1}) + \frac{1}{2}\log\det(WW^T + \Psi). \quad (107)$$

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

$$\frac{\partial S}{\partial W} = (WW^T + \Psi)^{-1}W^T - (WW^T + \Psi)^{-1}S(WW^T + \Psi)^{-1}W = 0 \quad (108)$$

$$\frac{\partial S}{\partial \Psi} = (WW^T + \Psi)^{-1} - (WW^T + \Psi)^{-1}S(WW^T + \Psi)^{-1} = 0, \quad (109)$$

leading to the fixed-point equations:

Fixed-point equations for the Stein divergence (110)

$$W = S(WW^T + \Psi)^{-1}W \quad (111)$$

$$\Psi = \mathrm{diag}(\mathrm{S} - \mathrm{W_n}\mathrm{W_n^T}). \quad (112)$$

B.2 Equivalence of fixed points and convergence

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

Proposition 2. *The factor analysis parameters which maximize the marginal likelihood, that is, the solution to* (105) *satisfy the following fixed point equation:*

$$W_n = S_N(WW^T + \Psi)^{-1}W \quad (113)$$

$$\Psi = \mathrm{diag}(S_N - W_n W_n^T). \quad (114)$$

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

$$W_n = S_N\Psi^{-1}W(\mathbf{I}_p + M^{-1}W^T\Psi^{-1}S_N\Psi^{-1}W)^{-1} \quad (115)$$

$$\textit{where: } M = \mathbf{I}_p + W^T\Psi^{-1}W \quad (116)$$

$$\Psi = \mathrm{diag}(S_N - W_n M^{-1}W^T\Psi^{-1}S_N). \quad (117)$$

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

Proof. The derivation of the fixed point equation from the Stein divergence is well known and recalled in Appendix B.1. The proof for the equivalence of the fixed points is then straightforward, since the update for W can be rewritten as:

$$W = S_N(WW^T + \Psi)^{-1}W \quad (118)$$

$$= S_N\Psi^{-1}W(\mathbf{I}_p + W^T\Psi^{-1}W)^{-1} \text{ (From the Woodbury formula)} \quad (119)$$

$$= S_N\Psi^{-1}WM^{-1} \quad (120)$$

$$= S_N\Psi^{-1}W(\mathbf{I}_p + (S_N\Psi^{-1}WM^{-1})^T\Psi^{-1}W)^{-1}, \quad (121)$$

where we have replaced the term W^T in (119) by its development in (120).

$$= S_N \Psi^{-1} W (\mathbf{I}_p + M^{-1} W^T \Psi^{-1} S_N \Psi^{-1} W)^{-1} = W_n. \tag{122}$$

The update for Ψ can be rewritten as:

$$\Psi = \mathrm{diag}(\mathrm{S_N} - \mathrm{WW^T}) \tag{123}$$
$$= \mathrm{diag}(S_N - W(S_N \Psi^{-1} W M^{-1})^T) \tag{124}$$

where we have replaced the term W^T in (123) by its development in (120), leading to

$$\Psi = \mathrm{diag}(S_N - W M^{-1} W^T \Psi^{-1} S_N) \tag{125}$$
$$= \mathrm{diag}(S_N - W_n M^{-1} W^T \Psi^{-1} S_N). \tag{126}$$

□

C Factor analysis fixed points and singular value decomposition (SVD).

C.1 Solution of the factor analysis fixed point with SVD.

We consider in this section how factor analysis with maximum likelihood can be solved with SVD. This is a well known result derived from the chapter 21.2 of the book of Barber (2011). The fixed point equation for factor analysis with maximum likelihood gives:

$$W = S(WW^T + \Psi)^{-1} W \tag{127}$$
$$\Psi = \mathrm{diag}(\mathrm{S - W_n W_n^T}), \tag{128}$$

the first equation can be rewritten to let an eigenvalue equation appear:

$$W = S\Psi^{-1} W (W^T \Psi^{-1} W + \mathbf{I}_p)^{-1} \text{ (from the Woodbury formula)} \tag{129}$$
$$\iff$$
$$(\hat{W}^T \hat{W} + \mathbf{I}_p)\hat{W} = \hat{S}\hat{W} \tag{130}$$
$$\text{where } \hat{S} = \Psi^{-\frac{1}{2}} S \Psi^{-\frac{1}{2}} \text{ and } \hat{W} = \Psi^{-\frac{1}{2}} W$$
$$\iff$$
$$ULV^T(\mathbf{I}_p + VL^2V^T) = \hat{S}ULV^T \tag{131}$$
$$\text{where } W = ULV^T \text{ with } U \in \mathcal{M}(d \times p) \text{ and } VV^T = \mathbf{I}_p \text{ (thin SVD)}$$
$$\iff$$
$$U(\mathbf{I}_p + L^2) = \hat{S}U. \tag{132}$$

And finally the solution is $W = UL$ where the columns of U are the p largest eigenvectors of $\hat{S}$ associated to the eigenvalues $\lambda_{i=1,\ldots,p}$ and $L = \mathrm{diag}(\mathrm{l_{i=1,\ldots,d}})$ where l_i depends on these maximal eigenvalues:

$$l_i = \sqrt{\lambda_i - 1} \quad \text{if} \quad \lambda_i \geq 1 \tag{133}$$
$$l_i = 0 \quad \text{otherwise .} \tag{134}$$

Since $\hat{S}$ depends on Ψ, we must iterate between updating W and updating Ψ until convergence. If we use a simpler probabilistic principal component analysis (PPCA) model, i.e. if $\Psi = \sigma^2\mathbf{I}$, the updates simplify to:

$$W = UL' = U\mathrm{diag}(\mathrm{l}_\mathrm{i}') \tag{135}$$

$$\text{where } l_i' = \sqrt{\lambda_i - \sigma^2} \quad \text{if} \quad \lambda_i \geq \sigma^2 \tag{136}$$

$$\text{and } l_i' = 0 \text{ otherwise} \tag{137}$$

$$\sigma^2 = \frac{1}{d-p}\sum_{i=p+1}^{d} \lambda_i, \tag{138}$$

these updates are decoupled and converge in one iteration. PPCA is much faster than factor analysis at the cost of a less expressive factorization.

C.2 EM as an implicite eigen decomposition

The fixed point equation from the maximum likelihood is solved using a SVD decomposition of S_N. The equivalence with the EM fixed point suggest that the EM make implicitly a 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 σ to 0. This results was shown by Roweis (1998) for the fixed point EM with PPCA and is extended to the Stein divergence fixed point in the following Corollary.

Corollary 1 (of Prop. 2)**.** *The minimum of the Stein divergence for probabilistic principal component analysis (PPCA):*

$$\min_{W,\sigma} \quad D_S(S_N, WW^T + \sigma^2\mathbf{I}_d) = \min_{W,\sigma} \quad \mathrm{Tr}(S_N(WW^T + \sigma^2\mathbf{I}_d)^{-1}) + \log\det(WW^T + \sigma^2\mathbf{I}_d), \tag{139}$$

satisfies the MLE fixed-point equation on W:

$$W_n = S_N(WW^T + \sigma^2\mathbf{I}_d)^{-1}W = S_NW(\sigma^2\mathbf{I}_p + W^TW)^{-1} \underset{\sigma\to 0}{\to} S_NW(W^TW)^{-1}, \tag{140}$$

which is equivalent to the (batch) power method when $\sigma \to 0$, defined by:

$$w \leftarrow S_N\frac{w}{||w||^2} \qquad \textit{(given here in vectorial form).} \tag{141}$$

This MLE fixed-point update is asymptotically equivalent to the EM fixed-point update on W:

$$W_n = S_NW(\sigma^2\mathbf{I}_p + (\sigma^2\mathbf{I}_p + W^TW)^{-1}W^TS_NW)^{-1} \underset{\sigma\to 0}{\to} S_NW(W^TS_NW)^{-1}W^TW, \tag{142}$$

which is equivalent to the EM-PCA method (Roweis, 1998) when $\sigma \to 0$, defined by :

$$X = (W^TW)^{-1}W^TY \tag{143}$$

$$W_n = YX^T(XX^T)^{-1} \textit{ where } Y \textit{ is defined such that } S_N = YY^T. \tag{144}$$

Proof. The proof is direct. □

We may also relate these updates with the Oja flow (Oja, 1982) given by :

$$\dot{W} = (\mathbf{I}_d - WW^T)S_N W, \tag{145}$$

which gives the stationary equation:

$$S_N W = WW^T S_N W \tag{146}$$

$$S_N W(W^T S_N W)^{-1} = W, \tag{147}$$

which is an unnormalized version of the EM-update:

$$S_N W(W^T S_N W)^{-1} W^T W = W. \tag{148}$$

D Proof of Lemma 3.

This lemma 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:

$$p(y|\theta) = m(y_t)\exp(\eta^T T(y) - A(\eta)), \tag{149}$$

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

$$-\frac{\partial^2 \ln p(y|\theta)}{\partial\theta^2} = -\frac{\partial h}{\partial\theta}\frac{\partial^2 \ln p(y|\theta)}{\partial h^2}\frac{\partial h}{\partial\theta}^T - \frac{\partial^2 h}{\partial\theta^2}\frac{\partial \ln p(y|\theta)}{\partial h} \tag{150}$$

$$= \frac{\partial h}{\partial\theta}Cov(y|\theta)\frac{\partial h}{\partial\theta}^T - \frac{\partial^2 h}{\partial\theta^2}(T(y) - m). \tag{151}$$

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

$$\mathbf{E}_y[-\frac{\partial^2 \ln p(y/\eta)}{\partial\theta^2}] = F(\theta) = \frac{\partial\eta}{\partial\theta}Cov(y|\theta)\frac{\partial\eta}{\partial\theta}^T. \tag{152}$$

And finally:

$$\mathrm{E}_\theta[F(\theta)] = \mathrm{E}_\theta[\frac{\partial h}{\partial\theta}\mathrm{Cov}(y|\theta)\frac{\partial h}{\partial\theta}^T]. \tag{153}$$

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 The L-RVGA in the general case: the updates equation for the mean

We have derived the updates for the covariance in the case of an exponential family distribution with a nonlinear model using the generalized Gauss Newton approximation, we derive now the updates for the mean. The gradient for an exponential family take the following form where we consider $\eta = h(\theta) = g(m)$ with m the mean of y:

$$\frac{\partial}{\partial \theta} \log p(y_t|\theta, x_t) = \frac{\partial}{\partial \theta}(\eta^T y_t - F(\eta)) \tag{154}$$

$$= y_t \frac{\partial h}{\partial \theta} - m \frac{\partial h}{\partial \theta} = (y_t - g^{-1} \circ h(\theta)) \frac{\partial h}{\partial \theta}, \tag{155}$$

where we have used the relation $\frac{\partial F}{\partial \eta} = m = g^{-1}(\eta)$.

As for the covariance updates we may approximate the expectation with sampling as follows:

$$\mathbf{E}_\theta[(y_t - g^{-1} \circ h(\theta)) \frac{\partial h}{\partial \theta}(\theta)] \tag{156}$$

$$\approx \frac{1}{K} \sum_{i=1}^{N} (y_t - g^{-1} \circ h(\theta)) \frac{\partial h}{\partial \theta}(\theta_i) = y_t \frac{1}{K} \sum_{i=1}^{N} d_i - \frac{1}{K} \sum_{i=1}^{N} h_i d_i \tag{157}$$

$$\text{where } h_i = g^{-1} \circ h(\theta_i) \text{ and } d_i = \frac{\partial h}{\partial \theta}(\theta_i). \tag{158}$$

Finally the RVGA explicite update equations:

$$P_t^{-1} = P_{t-1}^{-1} - \mathbf{E}_{\theta \sim \mathcal{N}(\mu_{t-1}, P_{t-1})}[\nabla_\theta^2 \log p(y_t|\theta, x_t)] \tag{159}$$

$$\mu_t = \mu_{t-1} + P_t \mathbf{E}_{\theta \sim \mathcal{N}(\mu_{t-1}, P_{t-1})}[\nabla_\theta \log p(y_t|\theta, x_t)], \tag{160}$$

can be written in the factorized form as:

$$P_t^{-1} \approx W_{t-1} W_{t-1}^T + \Psi_{t-1} + \frac{1}{K} \sum_{i=1}^{K} c_i c_i^T \approx W_t W_t^T + \Psi_t \tag{161}$$

$$\text{where} \quad c_i = \frac{\partial h}{\partial \theta}(\theta_i) \sqrt{\text{Cov}(y|\theta_i)^{-1}} \text{ and } \theta_i \sim \mathcal{N}(\mu_{t-1}, (W_{t-1} W_{t-1}^T + \Psi_{t-1})^{-1}) \tag{162}$$

$$\mu_t = \mu_{t-1} + (W_t W_t^T + \Psi_t)^{-1} (y_t \frac{1}{K} \sum_{i=1}^{N} d_i - \frac{1}{K} \sum_{i=1}^{N} h_i d_i) \tag{163}$$

$$\mu_t = \mu_{t-1} + \Psi_t^{-1} ((y_t \frac{1}{K} \sum_{i=1}^{N} d_i - \frac{1}{K} \sum_{i=1}^{N} h_i d_i) - W_t M_t^{-1} (W_t^T \Psi_t^{-1} (y_t \frac{1}{K} \sum_{i=1}^{N} d_i - \frac{1}{K} \sum_{i=1}^{N} h_i d_i))). \tag{164}$$

F Mirror prox

F.1 Mirror prox as an iterated Kalman filter

In this section we show that the iterated scheme defined equation (48) 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_η of natural parameter η, 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_η and the one-sample posterior $p(\theta|y_t) \propto p(y_t|\theta)q_{\eta_{t-1}}(\theta)$ writes:

$$\begin{aligned} &\arg\min_{\eta_t} \quad KL(q_{\eta_t}(\theta)|p(\theta|y_t)) && (165)\\ =&\arg\min_{\eta_t} \quad \mathbf{E}_{q_{\eta_t}}[-\log p(y_t|\theta)] + B_F(\eta_{t-1}, \eta_t), && (166)\end{aligned}$$

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

$$\nabla_{\eta_t}\mathbf{E}_{q_{\eta_t}}[-\log p(y_t|\theta)] + (\eta_t - \eta_{t-1})\nabla^2 F(\eta_t) = 0, \tag{167}$$

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

$$\begin{aligned} \eta_t &= \eta_{t-1} + (\nabla^2 F(\eta_t))^{-1}\nabla_\eta \mathbf{E}_{q_\eta}[\log p(y_t|\theta)](\eta_t) && (168)\\ &= \eta_{t-1} + \nabla_m \mathbf{E}_{q_m}[\log p(y_t|\theta)](m_t). && (169)\end{aligned}$$

If we consider the function $f(m) = \mathrm{E}_{q_m}[-\log p(y_t|\theta)]$ 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:

$$\begin{aligned} &\nabla F^*(m_t) = \nabla F^*(m_{t-1}) - \nabla_m f(m_t) \text{ (dual descent with step 1)} && (170)\\ &m_t = \nabla F(\eta_t). \text{ (projection)} && (171)\end{aligned}$$

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

Mirror prox

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

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 μ 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 μ, P using the chain rule:

$$\begin{aligned} \frac{\partial f}{\partial m_1} &= \frac{\partial f}{\partial \mu}\frac{\partial \mu}{\partial m_1} + \frac{\partial f}{\partial P}\frac{\partial P}{\partial m_1} = \frac{\partial f}{\partial \mu} - 2\frac{\partial f}{\partial P}\mu && (172)\\ \frac{\partial f}{\partial m_2} &= \frac{\partial f}{\partial \mu}\frac{\partial \mu}{\partial m_2} + \frac{\partial f}{\partial P}\frac{\partial P}{\partial m_2} = \frac{\partial f}{\partial P} && (173)\end{aligned}$$

A step of the Mirror prox update:

$$\nabla F^*(m_t) = \nabla F^*(m_{t-1}) - \nabla_m f(m_{t-1}) \tag{174}$$

$$\iff \eta_t = \eta_{t-1} - \nabla_m f(m_{t-1}), \tag{175}$$

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

$$\begin{pmatrix} P_t^{-1}\mu_t \\ -\frac{1}{2}P_t^{-1} \end{pmatrix} = \begin{pmatrix} P_{t-1}^{-1}\mu_{t-1} \\ -\frac{1}{2}P_{t-1}^{-1} \end{pmatrix} + \begin{pmatrix} \frac{\partial f}{\partial \mu}|_{\mu_{t-1},P_{t-1}} - 2\frac{\partial f}{\partial P}|_{\mu_{t-1},P_{t-1}}\mu_{t-1} \\ \frac{\partial f}{\partial P}|_{\mu_{t-1},P_{t-1}} \end{pmatrix} \tag{176}$$

$$\iff \begin{pmatrix} P_t^{-1}\mu_t \\ -\frac{1}{2}P_t^{-1} \end{pmatrix} = \begin{pmatrix} (P_{t-1}^{-1} - 2\frac{\partial f}{\partial P}|_{\mu_{t-1},P_{t-1}}\mu_{t-1} \\ -\frac{1}{2}P_{t-1}^{-1} \end{pmatrix} + \begin{pmatrix} \frac{\partial f}{\partial \mu}|_{\mu_{t-1},P_{t-1}} \\ \frac{\partial f}{\partial P}|_{\mu_{t-1},P_{t-1}} \end{pmatrix} \tag{177}$$

$$\iff \begin{pmatrix} P_t^{-1}\mu_t \\ -\frac{1}{2}P_t^{-1} \end{pmatrix} = \begin{pmatrix} P_t^{-1}\mu_{t-1} \\ -\frac{1}{2}P_{t-1}^{-1} \end{pmatrix} + \begin{pmatrix} \frac{\partial f}{\partial \mu}|_{\mu_{t-1},P_{t-1}} \\ \frac{\partial f}{\partial P}|_{\mu_{t-1},P_{t-1}} \end{pmatrix} \tag{178}$$

$$\iff \begin{pmatrix} P_t^{-1}\mu_t \\ -\frac{1}{2}P_t^{-1} \end{pmatrix} = \begin{pmatrix} P_t^{-1}\mu_{t-1} \\ -\frac{1}{2}P_{t-1}^{-1} \end{pmatrix} + \begin{pmatrix} -\mathrm{E}_{\theta\sim\mathcal{N}(\mu_{t-1},P_{t-1})}[\nabla_\theta \log p(y_t|\theta)] \\ \frac{1}{2}\mathrm{E}_{\theta\sim\mathcal{N}(\mu_{t-1},P_{t-1})}[\nabla^2_\theta \log p(y_t|\theta)] \end{pmatrix}. \tag{179}$$

The last derivation (179) comes from the Bonnet & Price formulas (Lin et al., 2019):

$$\nabla_\mu \mathcal{N}(\theta|\mu, P) = -\nabla_\theta \mathcal{N}(\theta|\mu, P) \tag{180}$$

$$\nabla_P \mathcal{N}(\theta|\mu, P) = \frac{1}{2}\nabla^2_\theta \mathcal{N}(\theta|\mu, P). \tag{181}$$

Rearranging terms and applying two times the update as in Mirror prox gives the iterated scheme defined in equation (48):

$$\hat{\mathbf{P}}_\mathbf{t}^{-1} = P_{t-1}^{-1} - \mathrm{E}_{\theta\sim\mathcal{N}(\mu_{t-1},P_{t-1})}[\nabla^2_\theta \log p(y_t|\theta)] \tag{182}$$

$$\hat{\mu}_\mathbf{t} = \mu_{t-1} + \hat{\mathbf{P}}_\mathbf{t}\mathrm{E}_{\theta\sim\mathcal{N}(\mu_{t-1},P_{t-1})}[\nabla_\theta \log p(y_t|\theta)] \tag{183}$$

$$\mathbf{P_t}^{-1} = P_{t-1}^{-1} - \mathrm{E}_{\theta\sim\mathcal{N}(\hat{\mu}_\mathbf{t},\hat{\mathbf{P}}_\mathbf{t})}[\nabla^2_\theta \log p(y_t|\theta)] \tag{184}$$

$$\mu_t = \mu_{t-1} = \mathbf{P_t}\mathrm{E}_{\theta\sim\mathcal{N}(\hat{\mu}_\mathbf{t},\hat{\mathbf{P}}_\mathbf{t})}[\nabla_\theta \log p(y_t|\theta)]. \tag{185}$$

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).

F.2 Joint convexity of the RVGA loss function in μ and C

Our Mirror prox algorithm operates on the function:

$$f(m) = \mathrm{E}_{q_m}[-\log p(Y|\theta)],$$

where $m = \begin{pmatrix} \mu \\ P + \mu\mu^T \end{pmatrix}$. Mirror prox was initially proposed with a step size parameter and several works (Bach and Levy, 2019, Babanezhad and Lacoste-Julien, 2020) have proven convergence results for particular tuning of this parameter when the function f is convex. Unfortunately this function is not convex in m. However, if p belongs to an exponential family with a GLM model, Challis and Barber (2013) have shown this function is jointly convex in μ and C where C is the Cholesky

decomposition of P, i.e. $P = CC^T$. The proof, in its shorter version proposed by M. K. Titsias (see Appendix added by Challis and Barber 2013), comes directly from the definition of the convexity. Indeed, if ϕ is convex ie $\phi(\alpha x + (1-\alpha)y) \leq \alpha\phi(x) + (1-\alpha)\phi(y)$, then the function $f(\mu, C) = \mathrm{E}_{z\sim\mathcal{N}(0,I)}[\phi(\mu + Cz)]$ is jointly convex in μ and C:

$$
\begin{aligned}
& f(\alpha\mu_1 + (1-\alpha)\mu_2, \alpha C_1 + (1-\alpha)C_2) && (186)\\
& = \mathrm{E}_{z\sim\mathcal{N}(0,I)}[\phi(\alpha\mu_1 + (1-\alpha)\mu_2 + (\alpha C_1 + (1-\alpha)C_2)z] && (187)\\
& \leq \mathrm{E}_{z\sim\mathcal{N}(0,I)}[\alpha\phi(\mu_1 + \alpha C_1 z) + (1-\alpha)\phi(\mu_2 + C_2 z)] && (188)\\
& = \alpha f(\mu_1, C_1) + (1-\alpha) f(\mu_2, C_2). && (189)
\end{aligned}
$$

Which proof that the function:

$$
f(\mu, C) = \mathrm{E}_{\mathcal{N}(\mu, CC^T)}[-\log p(Y|\theta)] = \mathrm{E}_{z\sim\mathcal{N}(0,I)}[-\log p(Y|C\theta + \mu)],
$$

is jointly convex in μ and C by the convexity of $-\log p$ for an exponential family with a GLM model.

F.3 The partial Mirror prox

In this section we asses the Mirror prox scheme in high dimension. Let us first assess the Mirror prox scheme in the case of logistic regression with a full covariance matrix (without factor analysis) and with analytical averaging as defined in equation (61). The Mirror prox scheme:

Mirror prox

$$
\begin{aligned}
\hat{\mathbf{P}}_\mathbf{t}^{-1} &= P_{t-1}^{-1} - \mathrm{E}_{\theta\sim\mathcal{N}(\mu_{t-1}, P_{t-1})}[\nabla_\theta^2 \log p(y_t|\theta)]\\
\hat{\mu}_\mathbf{t} &= \mu_{t-1} + \hat{\mathbf{P}}_\mathbf{t}\mathrm{E}_{\theta\sim\mathcal{N}(\mu_{t-1}, P_{t-1})}[\nabla_\theta \log p(y_t|\theta)]\\
\mathbf{P}_\mathbf{t}^{-1} &= P_{t-1}^{-1} - \mathrm{E}_{\theta\sim\mathcal{N}(\hat{\mu}_\mathbf{t}, \hat{\mathbf{P}}_\mathbf{t})}[\nabla_\theta^2 \log p(y_t|\theta)]\\
\mu_t &= \mu_{t-1} = \mathbf{P}_\mathbf{t}\mathrm{E}_{\theta\sim\mathcal{N}(\hat{\mu}_\mathbf{t}, \hat{\mathbf{P}}_\mathbf{t})}[\nabla_\theta \log p(y_t|\theta)],
\end{aligned}
\tag{190}
$$

is compared to the explicit scheme where we do only one step:

Explicit scheme

$$
\begin{aligned}
\mathbf{P}_\mathbf{t}^{-1} &= P_{t-1}^{-1} - \mathrm{E}_{\theta\sim\mathcal{N}(\mu_{t-1}, P_{t-1})}[\nabla_\theta^2 \log p(y_t|\theta)]\\
\mu_\mathbf{t} &= \mu_{t-1} + \mathbf{P}_\mathbf{t}\mathrm{E}_{\theta\sim\mathcal{N}(\mu_{t-1}, P_{t-1})}[\nabla_\theta \log p(y_t|\theta)],
\end{aligned}
\tag{191}
$$

and the implicit scheme where the averaged are computed on the unknown parameters:

Implicit scheme

$$
\begin{aligned}
\mathbf{P}_\mathbf{t}^{-1} &= P_{t-1}^{-1} - \mathrm{E}_{\theta\sim\mathcal{N}(\mu_t, P_t)}[\nabla_\theta^2 \log p(y_t|\theta)]\\
\mu_\mathbf{t} &= \mu_{t-1} + \mathbf{P}_\mathbf{t}\mathrm{E}_{\theta\sim\mathcal{N}(\mu_t, P_t)}[\nabla_\theta \log p(y_t|\theta)].
\end{aligned}
\tag{192}
$$

The Mirror prox is less biased than the explicit scheme and even the implicit scheme when the prior is sharp as shown in Figure 8.

We now assess the robustness of the Mirror prox scheme to an approximation of the covariance matrix and show experimentally how a partial Mirror prox can improve the convergence. Let us consider the following approximations:

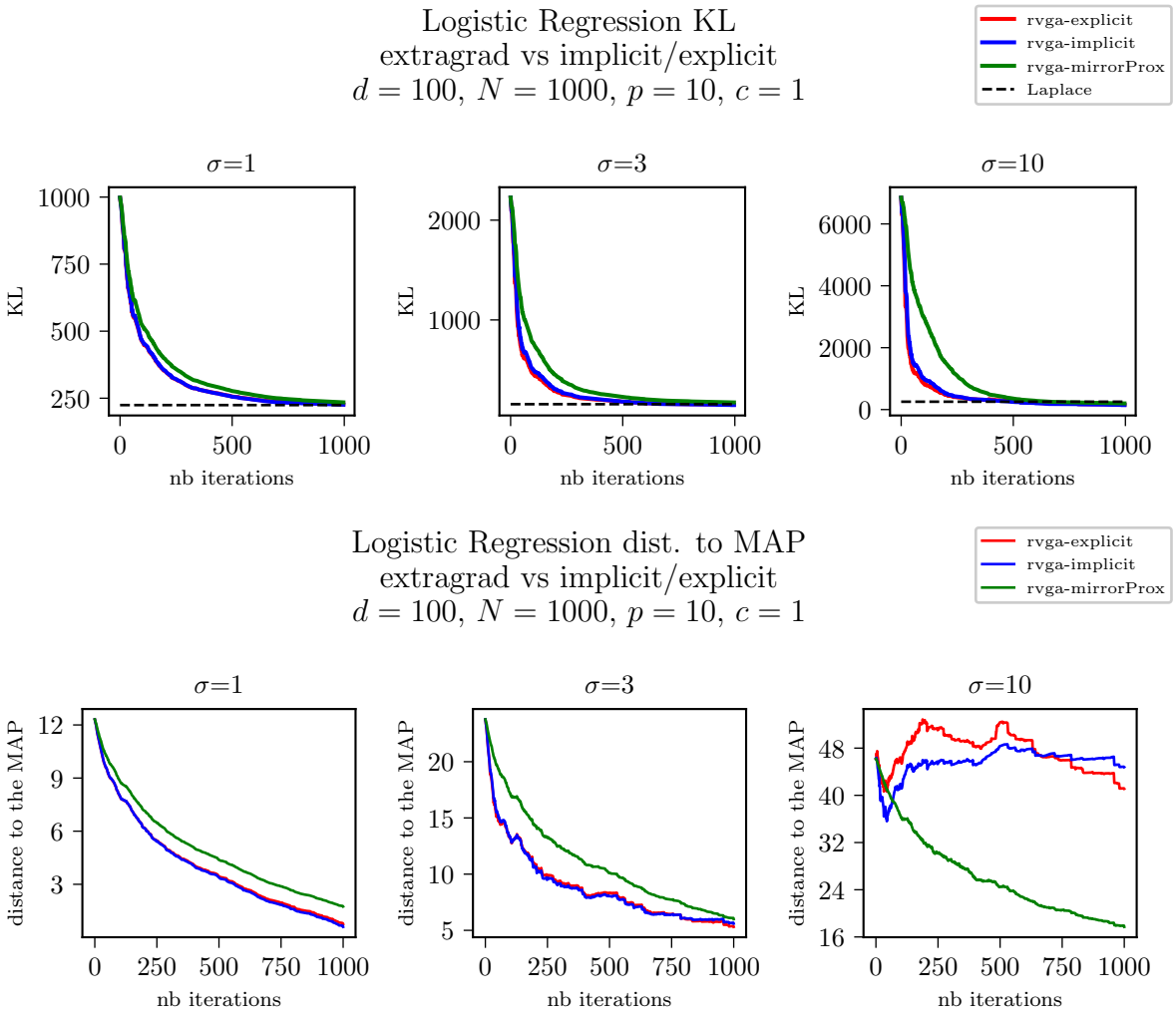

Figure 8: Left: Comparison of the Mirror prox with the implicit and explicit scheme for different priors : from smooth prior $\sigma_0 = 1$ (left plot) to sharp prior $\sigma_0 = 10$ (right plot). All versions decrease well in the KL divergence and converge to the same KL than the batch Laplace. Right: Comparison of the same algorithms but in term of bias. Considering the MAP is not so far from the mean of the logistic posterior for quite high values of σ_0, we may monitor the distance to the MAP to estimate the distance to the true mean (ie the bias). When the loss is sharper ($\sigma_0 = 10$), the explicit scheme appear strongly biased whereas the Mirror prox not. In this experiment, we have considered a full covariance matrix with analytical average. Note the posterior is known up to a constant so the KL are not normalized: relative values are meaningful but absolute value is not informative.

- **Mirror prox + FA** Mirror prox with approximation of the covariance matrix with factor analysis (FA) where the expectations are computed analytically as defined in equation (61).

- **Mirror prox + FA + sampling** Mirror prox with approximation of the covariance matrix with factor analysis (FA) where the expectations are computed with importance sampling as defined in equation (66).

- **partial Mirror prox + FA + sampling** the same as above but where we skip the extra update of the covariance:

$$
\begin{aligned}
&\textbf{partial Mirror prox} \\
&\hat{\mathbf{P}}_{\mathbf{t}}{}^{-1} = P_{t-1}^{-1} - \mathrm{E}_{\theta\sim\mathcal{N}(\mu_{t-1},P_{t-1})}[\nabla_\theta^2 \log p(y_t|\theta)] \\
&\hat{\mu}_{\mathbf{t}} = \mu_{t-1} + \hat{\mathbf{P}}_{\mathbf{t}} \mathrm{E}_{\theta\sim\mathcal{N}(\mu_{t-1},P_{t-1})}[\nabla_\theta \log p(y_t|\theta)] \\
&\mu_t = \mu_{t-1} + \hat{\mathbf{P}}_{\mathbf{t}} \mathrm{E}_{\theta\sim\mathcal{N}(\hat{\mu}_{\mathbf{t}},\hat{\mathbf{P}}_{\mathbf{t}})}[\nabla_\theta \log p(y_t|\theta)].
\end{aligned}
\tag{193}
$$

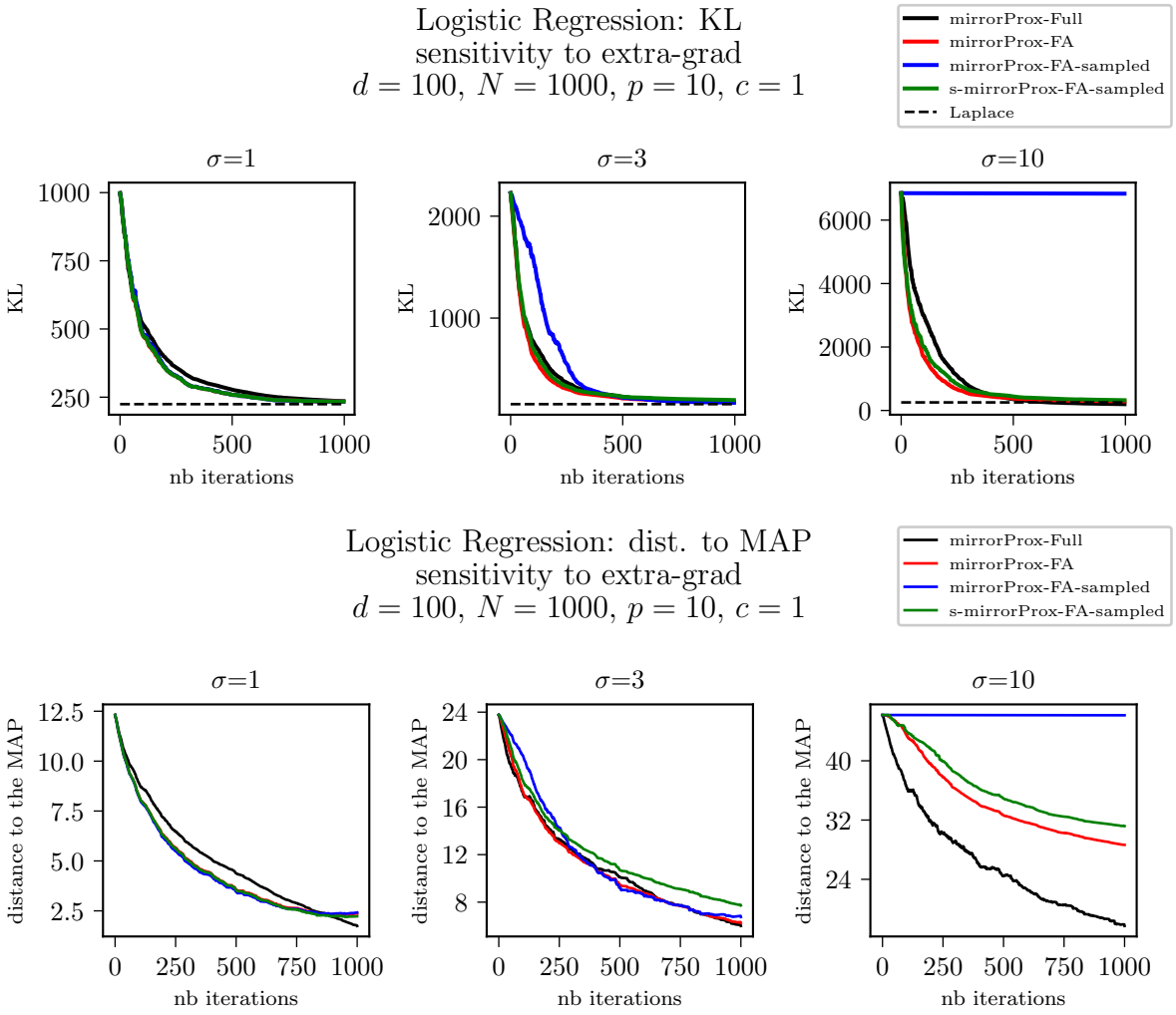

Figure 9: Comparison of the Mirror prox without approximation (MirrorProx-Full) with the three variants of approximated Mirror prox: with factor analysis only (MirrorProx-FA), with factor analysis and sampling (MirrorProx-FA-sampled) and with factor analysis and sampling but with a partial mirror prox step (s-MirrorProx-FA-sampled). Left: Comparison with the KL divergence from the true posterior. Right: Comparison in terms of bias. Considering the MAP is not so far from the mean of the logistic posterior for quite high values of σ_0, we may monitor the distance to the MAP to estimate the distance to the true mean (i.e. the bias). The Mirror prox does not anymore converge to the batch Laplace when the covariance is approximated and tends to diverge when we use sampling. Skipping the extra covariance update (partial Mirror prox) allows recovering a good convergence.