A probabilistic incremental proximal gradient method

Ömer Deniz Akyildiz, Émilie Chouzenoux, Víctor Elvira, Joaquín Míguez

***Abstract*—In this paper, we propose a probabilistic optimization method, named *probabilistic incremental proximal gradient* (PIPG) method, by developing a probabilistic interpretation of the incremental proximal gradient algorithm. We explicitly model the update rules of the incremental proximal gradient method and develop a systematic approach to propagate the uncertainty of the solution estimate over iterations. The PIPG algorithm takes the form of Bayesian filtering updates for a state-space model constructed by using the cost function. Our framework makes it possible to utilize well-known exact or approximate Bayesian filters, such as Kalman or extended Kalman filters, to solve large-scale regularized optimization problems.**

***Index Terms*—Probabilistic optimization, stochastic gradient, proximal algorithms, extended Kalman filtering**

I. Introduction

In this paper, we are interested in optimization problems of the form

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} f(\boldsymbol{\theta}) + g(\boldsymbol{\theta}), \tag{1}$$

with $f(\boldsymbol{\theta}) = \sum_{k=1}^n f_k(\boldsymbol{\theta})$ where, for $1 \leq k \leq n$, $f_k : \mathbb{R}^d \to \mathbb{R}$, are nonlinear least squares functions i.e., for $1 \leq k \leq n$, $f_k = \frac{1}{2}(y_k - h_k(\cdot))^2$, with $y_k \in \mathbb{R}$ and $h_k : \mathbb{R}^d \to \mathbb{R}$ a nonlinear differentiable mapping. Moreover, $g : \mathbb{R}^d \to \mathbb{R}$ is a twice-differentiable regularizer. Because classical optimization schemes may be inefficient to solve (1) when n is very large, stochastic or incremental optimization methods have gained a significant momentum. In particular, the stochastic gradient descent (SGD) [1] has become widely popular to solve such problems. At each SGD iteration, a mini-batch of component functions is randomly selected and a gradient step with respect to this mini-batch is performed. A number of variants of SGD have been since developed (see [2], [3] for a review).

The objective function in Eq. (1) has a sum structure. Therefore, it opens the door for more efficient algorithms than gradient methods, such as *proximal splitting* methods [4], [5]. In particular, the proximal gradient (PG) method minimizes a sum of two terms, one being smooth, by alternating gradient steps on the differentiable one and proximal update on the second, thereby exploiting fully the structure of the cost function. Naturally, stochastic extensions of proximal methods have become increasingly popular in the machine learning literature, see, e.g., [6]–[10]. The optimization method in consideration in this paper is known as the *incremental proximal gradient (IPG)* algorithm [7], and can be understood as an incremental version of the stochastic proximal gradient method [8], [9]. Similarly to its batch version PG, the IPG method would solve (1) by using the gradient of g and the proximal operator of f_k (or vice versa) at each iteration to move within the parameter space. Therefore, the IPG takes advantage of the structure of the cost function while staying computationally efficient for large n.

In this paper, we propose a probabilistic IPG (PIPG) method to solve the problem in Eq. (1). The PIPG algorithm reads as an approximate inference method in a probabilistic state-space model (SSM), tailored to the loss function. To be specific, it takes the form of an extended Kalman filter (EKF) to infer the hidden states of this SSM. This setting yields a probabilistic interpretation which enables the quantification of the uncertainty of the estimates at any time, extending our previous work [11] which only focused on the case $g = 0$. The posterior covariance matrix involved in PIPG updates plays the role of a variable-metric. Thus, another key advantage of PIPG is to provide an adaptive rule for the metric update within the IPG scheme. Note that the PIPG method is related to the class of *probabilistic numerical methods* (see, e.g., [12]–[14]), extending such methods for solving large-scale optimization problems. We mention [15] as a related work, that emphasizes the links between Kalman filtering and the online natural gradient method, which can be viewed as an SGD within a specific variable metric. In [16], connections between least-mean squares (LMS) and Kalman filters are exploited to propose a new algorithm. In [17], the author proposes some variance reduction strategies for SGD, relying on a Kalman interpretation. In contrast, in this work we take advantage of the structure of the cost function itself and we focus on the connection between Kalman and proximal methods.

The paper is organized as follows. In Section II, we briefly give background. In Section III, we introduce the new scheme and the update rules in detail. In Section IV, we demonstrate the performance of our method, on a ridge regression and a nonlinear sparse filter identification problem. We conclude with Section V.

Ö. D. Akyildiz is with the Dept. of Computer Science and Dept. of Statistics at University of Warwick and Alan Turing Institute, London, UK. Email: omer.akyildiz@warwick.ac.uk

É. Chouzenoux is with the Center for Visual Computing, INRIA Saclay, CentraleSupélec, Gif-sur-Yvette, France.

V. Elvira is with IMT Lille Douai & CRIStAL laboratory (UMR CNRS 9189), Villeneuve d'Ascq, France.

J. Míguez is with the Dept. of Signal Theory and Communications, Universidad Carlos III de Madrid, Leganés, Spain, 28912.

Ö. D. A. is funded by the Lloyds Register Foundation programme on Data Centric Engineering through the London Air Quality project and supported by The Alan Turing Institute for Data Science and AI under EPSRC grant EP/N510129/1. J. M. acknowledges the support of the Spanish *Agencia Estatal de Investigación* (TEC2015-69868-C2-1-R ADVENTURE) and the Office of Naval Research (N00014-19-1-2226). V. E. and É .C. acknowledge the support from the *Agence Nationale de la Recherche* of France under PISCES (ANR-17-CE40-0031-01) and MAJIC (ANR-17-CE40-0004-01) projects.

II. Background

Let us start by defining the proximal operator [18][1].

Definition 1. *The proximal operator of a convex, proper, lower semi-continuous function* $f : \mathbb{R}^d \to \mathbb{R}$ *within the metric induced by a symmetric, positive definite (SPD) matrix* $\mathbf{V}_0 \in \mathbb{R}^{d\times d}$ *is defined as,* $\operatorname{prox}_{f,\mathbf{V}_0}(\boldsymbol{\theta}_0) = \operatorname{argmin}_{\boldsymbol{\theta}\in\mathbb{R}^d} f(\boldsymbol{\theta}) + \frac{1}{2}\|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|^2_{2,\mathbf{V}_0}$ *where* $\|\boldsymbol{\theta}\|_{2,\mathbf{V}} := (\boldsymbol{\theta}^\top\mathbf{V}^{-1}\boldsymbol{\theta})^{1/2}$ *is the Mahalanobis distance.*

Let us now present the Kalman updates from [11] which aim at performing Bayesian inference in the case of the model $p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}; \overline{\boldsymbol{\theta}}_0, \mathbf{V}_0)$ and $p(y_k|\boldsymbol{\theta}) = \mathcal{N}(y_k; \mathbf{x}_k^\top\boldsymbol{\theta}, \gamma^{-1})$, where $\gamma > 0$, $\overline{\boldsymbol{\theta}}_0 \in \mathbb{R}^d$, $\mathbf{V}_0 \in \mathbb{R}^{d\times d}$ SPD, $\mathbf{x}_k \in \mathbb{R}^d$, for $k = 1, \dots, n$, are predefined values, and $(y_k, \boldsymbol{\theta})$ are random variables in $\mathbb{R}$ and $\mathbb{R}^d$, respectively. For this model, assuming that the inputs $\mathbf{x}_{1:k}$ are fixed and the likelihood factorizes as $p(y_{1:k}|\boldsymbol{\theta}) = \prod_{k=1}^n p(y_k|\boldsymbol{\theta})$ (i.e., the observations are conditionally independent), the mean and the covariance of the Gaussian posterior $p(\boldsymbol{\theta}|y_{1:k}) = \mathcal{N}(\boldsymbol{\theta}; \overline{\boldsymbol{\theta}}_k, \mathbf{V}_k)$ can be written as [11]

$$\overline{\boldsymbol{\theta}}_k = \overline{\boldsymbol{\theta}}_{k-1} + \frac{\mathbf{V}_{k-1}\mathbf{x}_k(y_k - \mathbf{x}_k^\top\overline{\boldsymbol{\theta}}_{k-1})}{\gamma^{-1} + \mathbf{x}_k^\top\mathbf{V}_{k-1}\mathbf{x}_k}, \tag{2}$$

$$\mathbf{V}_k = \mathbf{V}_{k-1} - \frac{\mathbf{V}_{k-1}\mathbf{x}_k\mathbf{x}_k^\top\mathbf{V}_{k-1}}{\gamma^{-1} + \mathbf{x}_k^\top\mathbf{V}_{k-1}\mathbf{x}_k}. \tag{3}$$

Note that at the last iteration, with $k = n$, the Gaussian posterior $p(\boldsymbol{\theta}|\mathbf{y}_{1:k})$ is perfectly computed with parameters given by Eqs. (2)-(3). The sequence $(\overline{\boldsymbol{\theta}}_k)_{1\le k\le n}$ turns out to be identical to the n first iterations of the incremental proximal method (IPM) recursion [6], [7] applied to Problem (1):

$$\overline{\boldsymbol{\theta}}_k = \operatorname{prox}_{\gamma f_k,\mathbf{V}_{k-1}}(\overline{\boldsymbol{\theta}}_{k-1}), \quad (\forall k \in \{1,\dots,n\}) \tag{4}$$

when $g = 0$ and

$$f_k(\boldsymbol{\theta}) = \frac{1}{2}(y_k - \mathbf{x}_k^\top\boldsymbol{\theta})^2, \quad (\forall\boldsymbol{\theta} \in \mathbb{R}^d) \tag{5}$$

for all $k \in \{1,\dots,n\}$ and $(\mathbf{V}_k)_{1\le k\le n}$ are specified as in (3)[2] (see Props. 4.2–4.4 in [20] for a proof). This viewpoint has been extended in [11] for nonlinear least squares, where

$$f_k(\boldsymbol{\theta}) = \frac{1}{2}(y_k - h_k(\boldsymbol{\theta}))^2, \quad \forall\boldsymbol{\theta} \in \mathbb{R}^d, \tag{6}$$

for all $k \in \{1,\dots,n\}$. In Eq. (6), each $h_k : \mathbb{R}^d \to \mathbb{R}$ is a differentiable function, possibly nonlinear. Thus, the IPM iteration of Eq. (4) may not be feasible in a closed form. One can implement the EKF for a model with prior $p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}; \overline{\boldsymbol{\theta}}_0, \mathbf{V}_0)$ and the likelihood $p(y_k|\boldsymbol{\theta}) = \mathcal{N}(y_k; h_k(\boldsymbol{\theta}), \gamma^{-1})$, by linearizing $(h_k)_{1\le k\le n}$. Denoting $\mathbf{d}_k = \nabla h_k(\boldsymbol{\theta}_{k-1})$, we obtain the update rules [11], [20]

$$\overline{\boldsymbol{\theta}}_k = \overline{\boldsymbol{\theta}}_{k-1} + \frac{\mathbf{V}_{k-1}\mathbf{d}_k(y_k - h_\kappa(\overline{\boldsymbol{\theta}}_{k-1}))}{\gamma^{-1} + \mathbf{d}_k^\top\mathbf{V}_{k-1}\mathbf{d}_k},$$

$$\mathbf{V}_k = \mathbf{V}_{k-1} - \frac{\mathbf{V}_{k-1}\mathbf{d}_k\mathbf{d}_k^\top\mathbf{V}_{k-1}}{\gamma^{-1} + \mathbf{d}_k^\top\mathbf{V}_{k-1}\mathbf{d}_k}.$$

for $k \in \{1,\dots,n\}$. Since the EKF is an approximate Bayesian scheme, multiple passes over the dataset can be performed.

III. A Probabilistic IPG method

We now focus on the resolution of the optimization problem in Eq. (1) when $g \neq 0$. The structure of the cost function suggests the use of the IPG iteration [6], [7]. We consider a variable-metric extension of the IPG. In particular, given (1), the n first iterations of the variable-metric IPG update read as

$$\overline{\boldsymbol{\theta}}_k = \operatorname{prox}_{\gamma f_k,\mathbf{V}_{k-1}}(\overline{\boldsymbol{\theta}}_{k-1} - \gamma\mathbf{V}_{k-1}\nabla g(\overline{\boldsymbol{\theta}}_{k-1})), \tag{7}$$

with $\overline{\boldsymbol{\theta}}_0 \in \mathbb{R}^d$, and $(\mathbf{V}_k)_{k\ge 0} \in \mathbb{R}^{d\times d}$ some predefined SPD matrices. The update (7) can be viewed as an incremental version of the batch variable-metric PG method that has been extensively studied recently in the optimization literature [21], [22]. In the sequel, we propose a probabilistic interpretation of the IPG which leads to a new update rule for the variable-metric matrices. We first consider the linear case (i.e., for quadratic f and g) for the sake of simplicity, since all computations are tractable and the inference can be performed in exact manner. Then we present our general version of the PIPG that encompasses a wider class of cost functions.

A. Linear-Quadratic case

Let us first assume that $(f_k)_{1\le k\le n}$ is defined as in (5) and

$$g(\boldsymbol{\theta}) = \frac{1}{2}\|\mathbf{A}\boldsymbol{\theta}\|_2^2 \quad (\forall\boldsymbol{\theta} \in \mathbb{R}^d), \tag{8}$$

with $\mathbf{A} \in \mathbb{R}^{m\times d}$, $m \ge 1$. Note that $\mathbf{A}$ is assumed to be known. Using (8), we can write (7) as

$$\widetilde{\boldsymbol{\theta}}_k = (\mathbf{I}_d - \gamma\mathbf{V}_{k-1}\mathbf{A}^\top\mathbf{A})\overline{\boldsymbol{\theta}}_{k-1}, \tag{9}$$

$$\overline{\boldsymbol{\theta}}_k = \operatorname{prox}_{\gamma f_k,\mathbf{V}_{k-1}}(\widetilde{\boldsymbol{\theta}}_k), \tag{10}$$

for $k = 1,\dots,n$. The key observation here is that Eqs. (9)–(10) can be seen as approximate (Kalman) filtering recursions [23]. To be specific, Eq. (9) can be seen as the analog to the prediction step within a Kalman filter. Similarly, the update (10) can be seen as a Bayesian update using (5), see Eq. (2) [11]. However, Eqs. (9)–(10) are different from a Kalman filter, where there would be an update of the covariance matrix between (9)–(10). Therefore, inspired by Eqs. (9)–(10), we propose the use of the following state-space model,

$$p(\boldsymbol{\theta}_0) = \mathcal{N}(\boldsymbol{\theta}_0; \overline{\boldsymbol{\theta}}_0, \mathbf{V}_0), \tag{11}$$

$$p(\boldsymbol{\theta}_k|\boldsymbol{\theta}_{k-1}) = \mathcal{N}(\boldsymbol{\theta}_k; \mathbf{M}_k\boldsymbol{\theta}_{k-1}, \mathbf{0}_{d\times d}), \tag{12}$$

$$p(y_k|\boldsymbol{\theta}_k) = \mathcal{N}(y_k; \mathbf{x}_k^\top\boldsymbol{\theta}_k, \gamma^{-1}), \tag{13}$$

where $\mathbf{0}_{d\times d}$, the zero-matrix in $\mathbb{R}^{d\times d}$, and

$$\mathbf{M}_k = (\mathbf{I}_d - \gamma\mathbf{V}_{k-1}\mathbf{A}^\top\mathbf{A}) \quad (\forall k \in \{1,\dots,n\}) \tag{14}$$

with $\mathbf{I}_d$ the identity matrix of $\mathbb{R}^d$. Now, assume that, the pair $(\overline{\boldsymbol{\theta}}_{k-1}, \mathbf{V}_{k-1})$ is given. We propose to apply filtering recursions for the model (11)–(13), which leads to the PIPG updates for the linear quadratic case. Recursions now consist of a predictive step of the mean and covariance

$$\widetilde{\boldsymbol{\theta}}_k = \mathbf{M}_k\overline{\boldsymbol{\theta}}_{k-1}, \tag{15}$$

$$\widetilde{\mathbf{V}}_k = \mathbf{M}_k\mathbf{V}_{k-1}\mathbf{M}_k^\top, \tag{16}$$

[1] See also `http://proximity-operator.net/`

[2] We note that, for this special case, the stochastic proximal point method for $\mathbf{V}_k = I_d$ also corresponds to the normalized LMS, see, e.g., Section 4.1 of [19].

respectively, and the update of the mean and covariance,

$$\overline{\boldsymbol{\theta}}_k = \widetilde{\boldsymbol{\theta}}_k + \frac{\widetilde{\mathbf{V}}_k \mathbf{x}_k (y_k - \mathbf{x}_k^\top \widetilde{\boldsymbol{\theta}}_k)}{\gamma^{-1} + \mathbf{x}_k^\top \widetilde{\mathbf{V}}_k \mathbf{x}_k}, \tag{17}$$

$$\mathbf{V}_k = \widetilde{\mathbf{V}}_k - \frac{\widetilde{\mathbf{V}}_k \mathbf{x}_k \mathbf{x}_k^\top \widetilde{\mathbf{V}}_k}{\gamma^{-1} + \mathbf{x}_k^\top \widetilde{\mathbf{V}}_k \mathbf{x}_k}, \tag{18}$$

respectively, with $(\mathbf{M}_k)_{1 \leq k \leq n}$ defined in (14). It is worth noting that, in Eqs. (9) and (10), a single $\mathbf{V}_{k-1}$ is used for both iterations. In the corresponding iterations in the proposed method, i.e., Eqs. (15) and (17), we make use of $\mathbf{V}_{k-1}$ and $\widetilde{\mathbf{V}}_k$, respectively. In this case, one pass over the dataset is enough since the posterior is exact for the model (11)–(13).

B. General case

In this section, we present the PIPG algorithm for the general nonlinear case. To be specific, we are going to focus on functions $(f_k)_{1 \leq k \leq n}$ taking the form (6). Moreover, we will consider a general function g that we assume to be twice differentiable. In this case, the variable-metric IPG update given in (7) does not usually yield analytically tractable computations. Moreover, the Kalman recursions, as we presented in the previous section, do not apply. To see this, first consider the mapping $m_{\mathbf{V}} : \mathbb{R}^d \mapsto \mathbb{R}^d$, where

$$m_{\mathbf{V}}(\overline{\boldsymbol{\theta}}) = \overline{\boldsymbol{\theta}} - \gamma \mathbf{V} \nabla g(\overline{\boldsymbol{\theta}}),$$

for some given $\overline{\boldsymbol{\theta}} \in \mathbb{R}^d$, $\mathbf{V} \in \mathbb{R}^{d \times d}$ SPD and $\gamma > 0$. Except when g is quadratic, the above mapping is nonlinear, making it impossible to propagate the uncertainty for the gradient step in (7) as it was done in (9). Moreover, when $(f_k)_{1 \leq k \leq n}$ are chosen as in (6), it may be complicated to realize the proximal step given in (7). To alleviate both problems, we can use the EKF [11], [23]. To this end, we build the model

$$p(\boldsymbol{\theta}_0) = \mathcal{N}(\boldsymbol{\theta}_0; \overline{\boldsymbol{\theta}}_0, \mathbf{V}_0), \tag{19}$$

$$p(\boldsymbol{\theta}_k | \boldsymbol{\theta}_{k-1}) = \mathcal{N}(\boldsymbol{\theta}_k; m_{\mathbf{V}_{k-1}}(\boldsymbol{\theta}_{k-1}), \mathbf{0}_{d \times d}), \tag{20}$$

$$p(y_k | \boldsymbol{\theta}_k) = \mathcal{N}(y_k; h_k(\boldsymbol{\theta}_k), \gamma^{-1}). \tag{21}$$

In order to apply the EKF in the model (19)–(21), which will lead to *the PIPG algorithm*, we need to linearize the transition model and the observation model. At iteration k, given $(\overline{\boldsymbol{\theta}}_{k-1}, \mathbf{V}_{k-1})$ pair, we define the transition matrix,

$$\mathbf{M}_k = \mathbf{I}_d - \gamma \mathbf{V}_{k-1} \nabla^2 g(\overline{\boldsymbol{\theta}}_{k-1}) \quad (\forall k \in \{1, \ldots, n\}),$$

with $\nabla^2 g$ the Hessian map of g. Finally, the PIPG updates can be computed: first the predicted mean and covariance

$$\widetilde{\boldsymbol{\theta}}_k = m_{\mathbf{V}_{k-1}}(\overline{\boldsymbol{\theta}}_{k-1}), \tag{22}$$

$$\widetilde{\mathbf{V}}_k = \mathbf{M}_k \mathbf{V}_{k-1} \mathbf{M}_k^\top + \mathbf{Q}, \tag{23}$$

respectively,[3] and then the updated mean and covariance

$$\overline{\boldsymbol{\theta}}_k = \widetilde{\boldsymbol{\theta}}_k + \frac{\widetilde{\mathbf{V}}_k \mathbf{d}_k (y_k - h_k(\widetilde{\boldsymbol{\theta}}_k))}{\gamma^{-1} + \mathbf{d}_k^\top \widetilde{\mathbf{V}}_k \mathbf{d}_k}, \tag{24}$$

$$\mathbf{V}_k = \widetilde{\mathbf{V}}_k - \frac{\widetilde{\mathbf{V}}_k \mathbf{d}_k \mathbf{d}_k^\top \widetilde{\mathbf{V}}_k}{\gamma^{-1} + \mathbf{d}_k^\top \widetilde{\mathbf{V}}_k \mathbf{d}_k}. \tag{25}$$

respectively, where $\mathbf{d}_k = \nabla h_k(\widetilde{\boldsymbol{\theta}}_k)$. The algorithm is iterated for $k = 1, \ldots, n$ and referred to as the PIPG method.

Remark 1. Note that Eqs. (22)–(25) are the most general recursions for our method. Like in the linear case presented in Section III-A, sometimes we can simplify the computations. For instance, if $m_{\mathbf{V}_{k-1}}(\cdot)$ yields a linear mapping for g while f is a nonlinear least squares loss as in (6), then Eqs. (22)–(23) simplify into (15)–(16). Similarly, when $m_{\mathbf{V}_{k-1}}(\cdot)$ is nonlinear, and f is quadratic as in (5), then Eqs. (24)–(25) simplify into (17)–(18).

Remark 2. Although the choice of metrics has been studied in the batch case [21], [24], no practical ways for choosing them are available in the incremental setting to the best of our knowledge. The PIPG scheme provides a natural recipe on how to update the metric matrices $(\mathbf{V}_k)_{1 \leq k \leq n}$ in the form of a sequence of posterior covariance matrices.

Remark 3. As mentioned earlier, in the linear and tractable case the PIPG updates given by (15)–(18), are guaranteed to provide, after $k = n$ iterations (i.e., after a single pass of the data), the exact mean and covariance parameters of the Gaussian posterior associated to the state-space model (11)-(13). However, the convergence analysis for the general recursions (22)–(25) (with inexact Kalman updates) would need further investigation that we leave for future work.

IV. Numerical results

In this section, we present two experiments in order to illustrate the performance of PIPG in the context described in Sections III-A and III-B.

A. Ridge regression

We consider first the linear-quadratic case, depicted in Section III-A. We set $\mathbf{A} = \sqrt{\lambda}\mathbf{I}_d$. Moreover, the sought signal $\boldsymbol{\theta}^\star \in \mathbb{R}^d$ is generated as the realization of a multivariate Gaussian variable using $d = 100$. We then simulated $n = 100,000$ noisy observations $y_k = \mathbf{x}_k^\top \boldsymbol{\theta}^\star + \eta_k$ with $\eta_k \sim \mathcal{N}(0, 1)$ for $k = 1, \ldots, n$. PIPG recursions (15)–(18) are implemented, for $n = k$ iterations and 40 step-size values γ withing the range $[0.005, 0.2]$. We also compare the results to the IPG obtained using (9)-(10) with $\mathbf{V}_k = \mathbf{I}_d$ for $k = 1, \ldots, n$. IPG was run with a decaying step-size of the form $\gamma / k^{0.51}$, for the same range of step-size values than PIPG. Note that running the IPG with a constant step-size causes the algorithm to diverge, therefore we do not show those results. For both methods, we access the data $(y_k, \mathbf{x}_k)_{1 \leq k \leq n}$ in a random order, hence the time dependency of $(\mathbf{x}_k)_{1 \leq k \leq n}$ is not affecting our results. We compute the relative mean squared error (RMSE) between the current estimate $\boldsymbol{\theta}_k$ and the true filter coefficient vector $\boldsymbol{\theta}^\star$ as $\mathrm{E}_k = \|\boldsymbol{\theta}_k - \boldsymbol{\theta}^\star\| / \|\boldsymbol{\theta}^\star\|$. The regularization parameter is set to $\lambda = 10^{-2}$ so as to minimize the final RMSE.

The results are displayed in Fig. 1(a). It can be seen that PIPG shows a stable performance with respect to the step-size value. In contrast, IPG appears to be very sensitive to both the step-size tuning and also the decay rate (which is not shown here). Moreover, for a wide range of step-size values,

[3] Note that, although the dynamical model (20) is deterministic (the process covariance matrix is zero), we have introduced $\mathbf{Q}$ in (23), a SPD matrix that accounts for the linearization error made by the EKF.

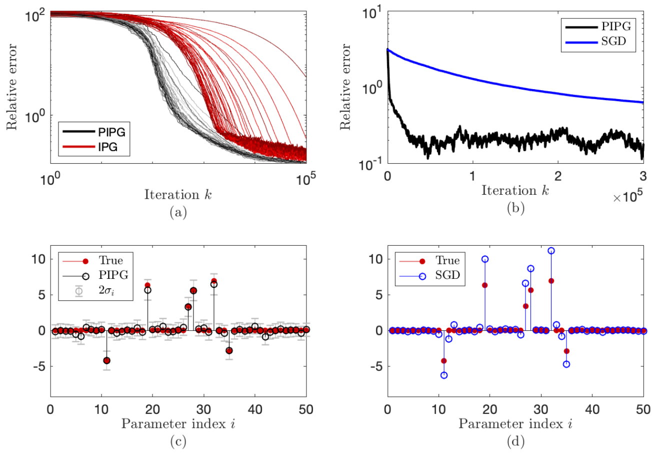

Fig. 1. (a) Ridge regression example, evolution of RMSE for different runs PIPG and IPG. (b-c-d) Sparse identification example. (b) Evolution of RMSE. (c): Posterior mean and uncertainty estimates, defined as $\pm 2\sigma_i$ for $i = 1, \ldots, d$, $d = 50$, with $(\sigma_i^2)_{1 \leq i \leq 50}$ the diagonal entries of the final posterior covariance matrix $\mathbf{V}_n$, after a single pass over the dataset. (d) SGD estimates after a single pass over the dataset.

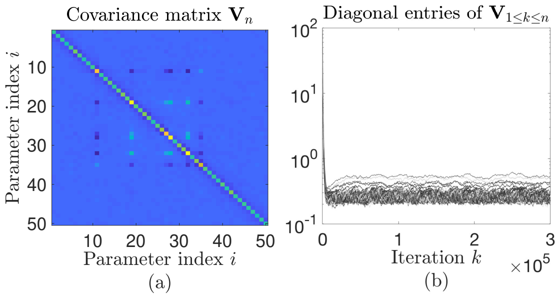

Fig. 2. The posterior covariance matrix $\mathbf{V}_n$ (a) and the diagonal entries of $(\mathbf{V}_k)_{1 \leq k \leq n}$ (b). From (a), it can be seen that the algorithm learns correlations between dimensions, which enables it to take more efficient steps. From (b), it can be observed that the diagonals of the sequence of covariance matrices converge to certain values quantifying the uncertainty of the final parameters.

PIPG requires less iterations than IPG to achieve minimal RMSE. Finally, PIPG provides an estimate of the covariance as an additional output, which can be particularly useful in practical applications that require an uncertainty quantification in the solution (e.g., biomedical data processing, financial data analytics).

B. Sparse nonlinear regression

Let us now apply the proposed method on the more challenging problem of sparse system identification [25], [26] under nonlinear observation model. Given a real-valued discrete-time input signal $\left(x_k\right)_{k \in \mathbb{Z}}$, the output of the system at time $k \in \{1, \ldots, n\}$ is defined as $y_k = h(\mathbf{x}_k^\top \boldsymbol{\theta}) + w_k$, where $\mathbf{x}_k = [x_{k-d+1}, \ldots, x_k]^\top \in \mathbb{R}^d$ (assuming circulant boundaries) and $w_k \sim \mathcal{N}(0, \gamma^{-1})$ are i.i.d. measurement noise samples, and $\boldsymbol{\theta} \in \mathbb{R}^d$ represents the unknown filter taps. A sigmoid nonlinearity $h(u) = 1/(1 + \exp(-u))$ for $u \in \mathbb{R}^d$ is introduced in the system response, modeling for instance some saturation of the sensor. We set the input signal $(x_k)_{k \in \mathbb{Z}}$ as in [27], $x_k = a x_{k-1} + \eta_k$, with $a = 0.8$, $\eta_k \sim \mathcal{N}(0, 1)$ and $x_0 \sim \mathcal{N}(0, 1)$. We run the PIPG recursions (22)–(25) from Section III-B, where we set $h_k(\boldsymbol{\theta}) = h(\mathbf{x}_k^\top \boldsymbol{\theta})$ for every $\boldsymbol{\theta} \in \mathbb{R}^d$, and the regularization function g is chosen as smoothed $\ell_2 - \ell_1$ regularization function [28] i.e., $g(\boldsymbol{\theta}) = \lambda \left(\sum_{i=1}^d \left(1 + \theta_i^2/\delta^2\right)^{1/2} - 1\right)$ with $\lambda > 0$ and $\delta > 0$ the smoothing parameter. Such regularizer allows to promote sparsity, as when $\delta \to 0$, the ℓ_1 norm is obtained. The measurement noise variance is $\gamma^{-1} = 1$. Note that the parameter γ is also the step-size in the proposed method, as we will discuss below. The filter length is $d = 50$ and the output of the system is observed at every time $k \in \{1, \ldots, n\}$ with $n = 300,000$. Regularization parameters are set manually to $(\lambda, \delta) = (10^{-5}, 0.1)$ so as to reach the best performance in terms of RMSE. We initialize the PIPG algorithm with a prior distribution with large uncertainty, namely $\mathbf{V}_0 = v_0 \mathbf{I}_d$, where $v_0 = 100$. The process noise covariance matrix, which models the linearization errors in our method, is chosen as $\mathbf{Q} = q\mathbf{I}_d$ with $q = 10^{-4}$. We set $\gamma = 1$, accordingly with the noise model. Note that, in general, γ is an unknown parameter that is to be set by the user depending on the approximate noise level. For comparison, we implement a stochastic gradient descent (SGD) with learning rate $\gamma_k^{\text{sgd}} = \frac{\alpha_0}{1+\alpha_1 k}$ for $k = \{1, \ldots, n\}$, where $\alpha_0 = 1$ and $\alpha_1 = 10^{-4}$ which are chosen to reach an optimal decrease. Note that, for this model, it is not possible to implement the IPG since $(f_k)_{1 \leq k \leq n}$ are not easily proximable.

Fig. 1(b) displaying RMSE evolution for both algorithms, shows that the PIPG method reaches stability in a reduced number of iterations, compared to the SGD, which is a significant practical advantage when one has a limited accessibility to the dataset. From Fig. 1(c)–(d), it can be seen that the PIPG method in Fig. 1(c) provides a better estimate together with the uncertainty bars $(2\sigma_i)_{1 \leq i \leq d}$. A great feature of PIPG is to provide estimates for the covariance matrix, which provides the uncertainty quantification on the parameters. The behavior of the entries of $(\mathbf{V}_k)_{1 \leq k \leq n}$ can be seen from Fig. 2, along with some comments. Let us remark that the computation of this matrix of dimension $d \times d$ implies an increase in computational complexity, as PIPG scales as $\mathcal{O}(d^2)$ while SGD scales as $\mathcal{O}(d)$.

V. Conclusions

We have proposed a probabilistic incremental optimization method which quantifies and propagates the uncertainty over its estimates. In the case of a regularized non-linear least squares, we have reinterpreted the classical IPG method as an approximate inference method in a state-space model. The extension of IPG to the probabilistic setting enables us to provide quantification of the uncertainties inherent in the numerical problem or caused by modeling errors. Our probabilistic interpretation also allows the use of accelerated variable metric updates, whose metric matrices are derived in an automatic and well-defined way. Future investigations will be devoted to the analysis of the convergence of the PIPG iterates, and the reduction of its complexity by means of suitable approximations.

References

[1] H. Robbins and S. Monro, "A stochastic approximation method," *Annals of Mathematical Statistics*, vol. 22, pp. 400–407, 1951.

[2] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *arXiv:1606.04838*, 2016.

[3] M. Pereyra, P. Schniter, E. Chouzenoux, J.-C. Pesquet, J.-Y. Tourneret, A. Hero, and S. McLaughlin, "A survey of stochastic simulation and optimization methods in signal processing," *IEEE J. Sel. Top. Signal Process.*, vol. 10, no. 2, pp. 224–241, Mar. 2016.

[4] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-point algorithms for inverse problems in science and engineering*. Springer, 2011, pp. 185–212.

[5] N. Parikh, S. Boyd *et al.*, "Proximal algorithms," *Foundations and Trends® in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[6] D. P. Bertsekas, "Incremental gradient, subgradient, and proximal methods for convex optimization: A survey," *Optimization for Machine Learning*, vol. 2010, pp. 1–38, 2011.

[7] ——, "Incremental proximal methods for large scale convex optimization," *Mathematical programming*, vol. 129, no. 2, pp. 163–195, 2011.

[8] L. Rosasco, S. Villa, and B. C. Vũ, "Convergence of stochastic proximal gradient algorithm," *arXiv preprint arXiv:1403.5074*, 2014.

[9] Y. F. Atchadé, G. Fort, and E. Moulines, "On perturbed proximal gradient algorithms," *J. Mach. Learn. Res*, vol. 18, no. 1, pp. 310–342, 2017.

[10] P. L. Combettes and J.-C. Pesquet, "Stochastic approximations and perturbations in forward-backward splitting for monotone operators," *Pure and Applied Functional Analysis*, vol. 1, no. 1, pp. 13–37, 2016.

[11] O. D. Akyildiz, V. Elvira, and J. Miguez, "The Incremental Proximal Method: A Probabilistic Perspective," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, Canada, April 2018.

[12] P. Hennig, M. A. Osborne, and M. Girolami, "Probabilistic numerics and uncertainty in computations," in *Proc. R. Soc. A*, vol. 471, no. 2179. The Royal Society, 2015, p. 20150142.

[13] P. Diaconis, "Bayesian numerical analysis," *Statistical decision theory and related topics IV*, vol. 1, pp. 163–175, 1988.

[14] J. Cockayne, C. Oates, T. Sullivan, and M. Girolami, "Bayesian probabilistic numerical methods," *arXiv preprint arXiv:1702.03673*, 2017.

[15] Y. Ollivier, "Online natural gradient as a Kalman filter," *Electronic Journal of Statistics*, vol. 12, no. 2, pp. 2930–2961, 2018.

[16] J. Fernandez-Bes, V. Elvira, and S. Van Vaerenbergh, "A probabilistic least-mean-squares filter," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 2199–2203.

[17] J. Vuckovic, "Kalman gradient descent: Adaptive variance reduction in stochastic optimization," *arXiv preprint arXiv:1810.12273*, 2018.

[18] H. H. Bauschke and P. L. Combettes, *Convex analysis and monotone operator theory in Hilbert spaces*. Springer, 2011, vol. 408.

[19] P. Toulis, T. Horel, and E. M. Airoldi, "Stable robbins-monro approximations through stochastic proximal updates," *arXiv preprint arXiv:1510.00967*, 2015.

[20] Ö. D. Akyildiz, "Sequential and adaptive Bayesian computation for inference and optimization," Ph.D. dissertation, Universidad Carlos III de Madrid, March 2019, can be accessed from: http://akyildiz.me/works/thesis.pdf.

[21] E. Chouzenoux, J.-C. Pesquet, and A. Repetti, "Variable metric forward-backward algorithm for minimizing the sum of a differentiable function and a convex function," *J. Optim. Theory Appl.*, vol. 162, no. 1, pp. 107–132, Jul. 2014.

[22] P. L. Combettes and B. C. Vũ, "Variable metric forward-backward splitting with applications to monotone inclusions in duality," *Optimization*, vol. 63, no. 9, pp. 1289–1318, 2014.

[23] B. D. Anderson and J. B. Moore, *Optimal filtering*. Englewood Cliffs, N.J. Prentice Hall, 1979.

[24] S. Becker and J. Fadili, "A quasi-newton proximal splitting method," in *Neural Information Processing Systems (NIPS 2012)*, no. 25, Lake Tahoe, Nevada, Dec. 2012, pp. 2618–2626.

[25] P. S. Diniz, *Adaptive filtering*. Springer, New York, NY, 4th edition, 2013.

[26] D. Angelosante, J. A. Bazerque, and G. B. Giannakis, "Online adaptive estimation of sparse signals: Where rls meets the ℓ_1-norm," *IEEE Transactions on Signal Processing*, vol. 58, no. 7, pp. 3436–3447, 2010.

[27] Y. Chen, Y. Gu, and A. O. Hero, "Regularized least-mean-square algorithms," *arXiv preprint arXiv:1012.5066*, 2010.

[28] E. Chouzenoux and J.-C. Pesquet, "A stochastic majorize-minimize subspace algorithm for online penalized least squares estimation," *IEEE Transactions on Signal Processing*, vol. 65, no. 18, pp. 4770–4783, 2017.