

VARIABLE PROJECTION FOR NONSMOOTH PROBLEMS*

TRISTAN VAN LEEUWEN[†] AND ALEKSANDR Y. ARAVKIN[‡]

Abstract. Variable projection solves structured optimization problems by completely minimizing over a subset of the variables while iterating over the remaining variables. Over the past 30 years, the technique has been widely used, with empirical and theoretical results demonstrating both greater efficacy and greater stability compared to competing approaches. Classic examples have exploited closed-form projections and smoothness of the objective function. We extend the approach to problems that include nonsmooth terms, develop an inexact adaptive algorithm that solves projection subproblems inexactly by iterative methods, and analyze its computational complexity. Finally, we illustrate the effectiveness of the adaptive algorithm with numerical examples. Code to reproduce the examples is available at <https://github.com/TristanvanLeeuwen/VarProNS>.

Key words. variable projection, proximal gradient, inexact gradient

AMS subject classifications. 68Q25, 68R10, 68U05

DOI. 10.1137/20M1348650

1. Introduction. In this paper we consider finite-dimensional *separable* optimization problems of the form

$$(1.1) \quad \min_{x,y} f(x,y) + r_1(x) + r_2(y),$$

where f smoothly couples (x, y) but may be nonconvex, while r_1 and r_2 encode additional constraints or regularizers. We are particularly interested in the case where $f(x, \cdot) + r_2$ is strongly convex in y , so that fast solvers are available for optimizing over y for fixed x . These problems arise any time nonsmooth regularization or constraints are used to regularize certain difficult nonlinear inverse problems or regression problems. We give three motivating examples below.

1.1. Motivating examples.

Model calibration. Consider the nonlinear fitting problem

$$(1.2) \quad \min_{x,y} \|A(x)y - b\|^2,$$

where $A(x)$ defines a linear model with calibration parameters x and b denotes the data. Well-known examples include exponential data fitting and model calibration in inverse problems. Adding regularization terms immediately gives a problem of the form (1.1).

Machine learning. Trimming is a model-agnostic tool for guarding against outliers. Given any machine learning model that minimizes some objectives ℓ_i over a training set of m data points, we introduce m auxiliary parameters y that serve to

*Received by the editors June 29, 2020; accepted for publication (in revised form) February 19, 2021; published electronically May 13, 2021.

<https://doi.org/10.1137/20M1348650>

[†]Centrum Wiskunde & Informatica, Amsterdam, 1098 XG, The Netherlands (T.van.Leeuwen@cwi.nl).

[‡]Department of Applied Mathematics, University of Washington, Seattle, WA 98195 USA (saravkin@uw.edu).

distinguish inliers from outliers and solve

$$\min_{x,y} \sum_{i=1}^m y_i \ell_i(x) \quad \text{s.t.} \quad 0 \leq y_i \leq 1 \text{ for } i \in \{1, \dots, m\}, \quad \sum_{i=1}^m y_i = k,$$

where $k \leq m$ is the number of datapoints we want to fit. The functions ℓ_i can capture a wide range of machine learning models.

PDE-constrained optimization. Many PDE-constrained optimization problems in data assimilation, inverse problems, and optimal control can be cast as

$$\min_{x,y} \|Py - d\|^2 + \lambda \|A(x)y - q\|^2,$$

where y denotes the state of the system, P is the sampling operator, $A(x)y = q$ is the discretized PDE with coefficients y and source term q , and λ is a penalty parameter. Adding regularization terms or changing the data fidelity term gives a problem of the form (1.1).

1.2. Approach. The development of specialized algorithms for (1.1) goes back to the classic *variable projection* (VP) technique for *separable nonlinear least-squares problems* of the form (1.2), where the matrix-valued map $x \mapsto A(x)$ is smooth and the matrix $A(x)$ has full rank for each x . Early work on the topic, notably by [8], has found numerous applications in chemistry, mechanical systems, neural networks, and telecommunications. See the surveys of [9] and [10] and references therein.

The VP approach is based on eliminating the variable y , as for each fixed x , we have a closed-form solution

$$\bar{y}(x) = A(x)^\dagger b,$$

where $A(x)^\dagger$ denotes the Moore–Pensrose pseudo-inverse of $A(x)$. We can thus express (1.2) in reduced form as

$$(1.3) \quad \min_x \|(A(x)A(x)^\dagger - I)b\|^2,$$

which is a nonlinear least-squares problem. Note that $A(x)A(x)^\dagger - I$ is an orthogonal projection onto the null-space of $A(x)^T$; hence, the name *variable projection*.

It was shown by [8] that the Jacobian of $A(x)A(x)^\dagger b$ contains only partial derivatives of $A(x)$ w.r.t. x and does not include derivatives of $\bar{y}(x)$ w.r.t. x . Ruhe and Wedin [14] showed that when the Gauss–Newton method for (1.2) converges superlinearly, so do certain Gauss–Newton variants for (1.3). Numerical practice shows that the latter schemes actually outperform the former because of a better conditioning of the reduced problem.

The underlying principle of the VP method is much broader than the class of separable nonlinear least-squares problems. For example, [5, 2] consider the class of problems

$$(1.4) \quad \min_{x,y} f(x,y),$$

where f is a C^2 -smooth function; the classic VP problem (1.2) is a special case of (1.4). Although we generally do not have a closed-form expression for $\bar{y}(x)$, we define it as

$$\bar{y}(x) = \operatorname{argmin}_y f(x,y)$$

and express (1.4) using the projected function

$$(1.5) \quad \bar{f}(x) := f(x, \bar{y}(x)).$$

Projection in the broader context of (1.4) refers to *epigraphical projection* [12] or partial minimization of y . Under mild conditions, $\bar{f}(x)$ is C^2 -smooth as well, and its gradient is given by

$$(1.6) \quad \nabla \bar{f}(x) = \nabla_x f(x, y) \Big|_{y=\bar{y}(x)};$$

i.e., it is the gradient of f w.r.t. x , evaluated at $\bar{y}(x)$ [5]. Again, we do not need to compute any sensitivities of $\bar{y}(x)$ w.r.t. x . This is seen by formally computing the gradient of \bar{f} using the chain rule:

$$(1.7) \quad \nabla \bar{f}(x) = \left(\nabla_x f(x, y) + \nabla_y f(x, y) \cdot \nabla_x \bar{y}(x) \right) \Big|_{y=\bar{y}(x)}.$$

Since $\bar{y}(x)$ is a minimizer of $f(x, y)$, it satisfies $\nabla_y f(x, \bar{y}(x)) = 0$, and the second term vanishes. Similarly, the Hessian of \bar{f} is the Schur complement of $\nabla_{yy}^2 f$ of the full Hessian of f [14]:

$$(1.8) \quad \nabla^2 \bar{f}(x) = \left(\nabla_{xx}^2 f(x, y) - \nabla_{xy}^2 f(x, y) (\nabla_{yy}^2 f(x, y))^{-1} \nabla_{yx}^2 f(x, y) \right) \Big|_{y=\bar{y}(x)}.$$

It follows that a local minimizer, \bar{x} , of \bar{f} together with $\bar{y}(\bar{x})$ constitute a local minimizer of f . An interlacing property of the eigenvalues of the Schur complement can be used to show that the reduced problem has a smaller condition number than the original problem [17]. The expression for the derivative furthermore suggests that we can approximate the gradient of \bar{f} when $\bar{y}(x)$ is known only approximately by ignoring the second term.

We may extend this approach to solve problems of the form (1.1) by including r_2 in the computation of $\bar{y}(x)$ and using an appropriate algorithm to minimize $\bar{f} + r_1$. Define the proximity operator for any function g as

$$\text{prox}_{\alpha g}(z) = \underset{x}{\operatorname{argmin}} \frac{1}{2\alpha} \|x - z\|^2 + g(x),$$

where $\alpha > 0$ is any scaling factor or step size. We can now view the entire approach as proximal-gradient descent on the projected function \bar{f} ,

$$(1.9) \quad x_{k+1} = \underset{\alpha r_1}{\operatorname{prox}}(x_k - \alpha \nabla \bar{f}(x_k)),$$

where α is an appropriate step and $\nabla \bar{f}$ is computed using (1.6). This gives rise to the following prototype Algorithm 1.1.

Naively, this approach can be applied to nonsmooth problems; however, it is not immediately obvious that it is guaranteed to converge. In particular, the resulting reduced objective \bar{f} may not be smooth, and hence the gradient formula (1.6) may not be valid. This is illustrated in the following example.

Example. To illustrate the possibilities and limitation for extending the VP approach to nonsmooth functions, consider the following functions:

$$\begin{aligned} F_1(x, y) &= \frac{1}{2}(x - y)^2 + \frac{1}{2}y^2, \\ F_2(x, y) &= \frac{1}{2}(x - y)^2 + |y|, \\ F_3(x, y) &= \frac{1}{2}(x - y)^2 + \delta_{[-1,1]}(y), \\ F_4(x, y) &= |x - y| + x^2|y|. \end{aligned}$$

Algorithm 1.1 Prototype VP algorithm for solving (1.1).

Require: Initial iterate, x_0 , Lipschitz constant, \bar{L} , of $\nabla \bar{f}$

$$\alpha = 1/\bar{L}$$

$$k = 0$$

while not converged **do**

$$\bar{y}_{k+1} = \operatorname{argmin}_y f(x_k, y) + r_2(y)$$

$$x_{k+1} = \operatorname{prox}_{\alpha r_1}(x_k - \alpha \nabla_x f(x_k, \bar{y}_{k+1}))$$

$$k = k + 1$$

end while

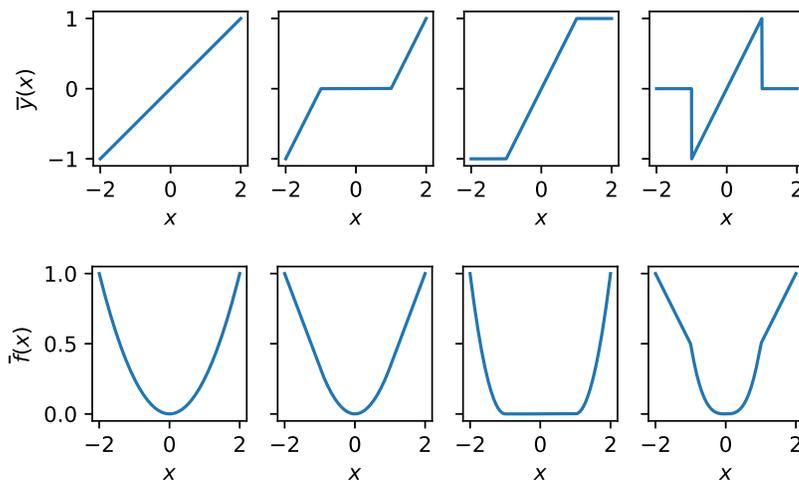


FIG. 1.1. Optimal solution(s) $\bar{y}(x)$ and projected function $\bar{f}(x)$ for functions F_1 – F_4 . For cases 1–3, \bar{y} is continuous (but not necessarily smooth), leading to a smooth projected function. For F_4 , \bar{y} is not continuous (at $x = \pm 1$, the solution is not unique), leading to a nonsmooth projected function.

The corresponding $\bar{y}(x)$ and $\bar{f}(x)$ are shown in Figure 1.1. For cases 1–3, \bar{y} is continuous (but not smooth), leading to a *smooth* projected function. For F_4 , \bar{y} is not continuous (at $x = \pm 1$, the solution is not unique), leading to a nonsmooth projected function. Continuity of \bar{y} thus appears to be important rather than smoothness of $F(x, y)$ in y . We therefore restrict our attention to problems of the form (1.1) that are *strongly* convex in y for all x of interest.

The goal of this paper is to extend the variable projection technique to problems of the form (1.1) with *nonsmooth* regularization terms, which arise in high-dimensional statistics, signal processing, and many machine learning problems; sparse regularization and simple constraints are frequently used in this setting.

1.3. Contributions and outline. Our contributions are as follows:

1. sufficient conditions under which \bar{f} is smooth and its gradient can be evaluated by (1.6);
2. development and analysis of an inexact adaptive version of Algorithm 1.1 based on inexact evaluations of $\bar{y}(x)$.

In section 2, we develop derivative formulas for the value function (1.5) and design an inexact version of Algorithm 1.1. In section 3, we present a few case studies. Conclusions complete the paper.

2. Derivative formulas and inexact VP. In this section, we present derivative formulas and develop the approaches briefly described in the introduction. The proofs of the following statements are found in the appendices. By $\|\cdot\|$, we denote the Euclidean norm. We make the following blanket assumptions on f and $F(x, y) = f(x, y) + r_2(y)$:

A1 $\nabla_x f$ and $\nabla_y f$ exist and are Lipschitz continuous for all (x, y) :

$$\begin{aligned} \|\nabla_x f(x, y) - \nabla_x f(x', y)\| &\leq L_{xx}\|x - x'\|, \\ \|\nabla_x f(x, y) - \nabla_x f(x, y')\| &\leq L_{xy}\|y - y'\|, \\ \|\nabla_y f(x, y) - \nabla_y f(x', y)\| &\leq L_{yx}\|x - x'\|, \\ \|\nabla_y f(x, y) - \nabla_y f(x, y')\| &\leq L_{yy}\|y - y'\|. \end{aligned}$$

A2 $F(x, y)$ is μ -strongly convex in y for all x .

2.1. Derivative formulas. We first establish Lipschitz continuity of the solution \bar{y} w.r.t. x in the following lemma.

LEMMA 2.1. *Let F satisfy assumptions (A1)–(A2). Then $\bar{y}(x) = \operatorname{argmin}_y F(x, y)$ is Lipschitz continuous as a function of x :*

$$\|\bar{y}(x) - \bar{y}(x')\| \leq (L_{yx}/\mu)\|x - x'\|.$$

Next, we establish that the naive derivative formula (1.7) holds under the aforementioned assumptions.

THEOREM 2.2. *Let F satisfy assumptions (A1)–(A2), and define $\bar{f}(x) = \min_y F(x, y)$. The gradient of the projected function is then given by $\nabla \bar{f}(x) = \nabla_x f(x, \bar{y}(x))$, with $\bar{y}(x) = \operatorname{argmin}_y F(x, y)$.*

Finally, we establish Lipschitz continuity of the gradient of the projected function.

COROLLARY 2.3. *The gradient of \bar{f} is Lipschitz continuous,*

$$\|\nabla \bar{f}(x) - \nabla \bar{f}(x')\| \leq \bar{L}\|x - x'\|,$$

where

$$\bar{L} = L_{xx} + L_{xy}L_{yx}/\mu.$$

Remark 2.4. Note that the bound in Corollary 2.3 is consistent with the expression for the Hessian in (1.8) in the smooth case.

These results immediately establish the convergence of Algorithm 1.1 to a stationary point for a broad class of problems of the form (1.1).

2.2. Inexactness. In many applications, we do not have a closed-form expression for \bar{y} , and it must be computed with an iterative scheme. An obvious choice is to use a proximal gradient method

$$y_{l+1} = \operatorname{prox}_{\beta r_2}(y_l - \beta \nabla_y f(x, y_l)),$$

with $\beta \in (0, 2/L_{yy})$. The resulting approximation, $\tilde{y}(x)$, of $\bar{y}(x)$ yields an approximation of the gradient of \bar{f} with error

$$\|\nabla_x f(x, \bar{y}(x)) - \nabla_x f(x, \tilde{y}(x))\| \leq L_{xy}\|\bar{y}(x) - \tilde{y}(x)\|.$$

This gives rise to an inexact counterpart of (1.9),

$$(2.1) \quad x_{k+1} = \operatorname{prox}_{\alpha r_1}(x_k - \alpha \nabla_x f(x_k, \tilde{y}(x_k))),$$

with $\alpha \in (0, 2/\bar{L})$. We denote the error in the gradient as $e_k = \nabla_x f(x_k, \tilde{y}(x_k)) - \nabla_x f(x_k, \bar{y}(x_k))$. We can immediately bound this error as

$$(2.2) \quad \|e_k\| \leq L_{xy} \|\bar{y}(x_k) - \tilde{y}(x_k)\|.$$

Conditions under which such iterations converge in case \bar{f} is (strongly) convex have been well studied [7, 15]. The basic gist of these results is that convergence of (2.1) can be ensured when the error decays sufficiently fast with k .

We did not find any results for general \bar{f} in the literature and therefore establish convergence of the inexact prox-gradient method for general Lipschitz-smooth \bar{f} in the following theorem.

THEOREM 2.5 (convergence of inexact proximal gradient, general case). *Denote*

$$\bar{F}(x) := \bar{f}(x) + r_1(x),$$

and let \bar{L} be the Lipschitz constant of $\nabla \bar{f}$. The iteration

$$(2.3) \quad x_{k+1} = \operatorname{prox}_{\alpha r_1}(x_k - \alpha(\nabla \bar{f}(x_k) + e_k)),$$

using step size $\alpha = 1/\bar{L}$ and with errors obeying $\|e_k\| \leq C\|x_{k+1} - x_k\|$ with $C < \bar{L}/2$, produces iterates for which

$$\min_{k \in \{0, 1, \dots, n-1\}} \|x_{k+1} - x_k\| \leq A \sqrt{\frac{\bar{F}(x_0) - \bar{F}(x_*)}{n}},$$

with $A = \sqrt{\frac{2}{\bar{L} - 2C}}$.

The proof of this theorem along with supporting lemmas are presented in Appendix D. This result immediately establishes convergence of (2.1) for problems satisfying assumptions A1–A2 when the errors in the gradient obey $\|e_k\| < (\bar{L}/2)\|x_{k+1} - x_k\|$. We discuss how to ensure this in practice in a subsequent section.

Stronger statements on the rate of convergence can be made by making stronger assumptions about f . When \bar{f} is convex (i.e., when f is jointly convex in (x, y)), a *sublinear* convergence rate of $\mathcal{O}(1/k)$ can be ensured when $\|e_k\| = \mathcal{O}(1/k^{1+\delta})$ for any $\delta > 0$ [15, Prop. 1]. A *linear* convergence for *strongly* convex \bar{f} can be ensured when $\|e_k\| = \mathcal{O}(\gamma^k)$ for any $\gamma < 1$ [15, Prop. 3].

2.3. Asymptotic complexity. In this section, we analyze the asymptotic complexity of finding an $\bar{\epsilon}$ -optimal estimate \tilde{x} of the solution to (1.1) which satisfies

$$|\bar{f}(\tilde{x}) - \bar{f}(\bar{x})| \leq \bar{\epsilon}.$$

We measure the complexity in terms of the total number of *inner* iterations required to achieve this. To usefully analyze the asymptotic complexity, we assume that \bar{f} is (strongly) convex.

We first note that we can produce an ϵ -optimal estimate of \bar{y} for which $\|\tilde{y} - \bar{y}\| \leq \epsilon$ in $\mathcal{O}(\log 1/\epsilon)$ *inner* iterations. This follows directly from linear convergence of the proximal gradient method for strongly convex problems. This also implies that we can compute an approximation of the gradient of $\nabla \bar{f}$ with error bounded by $L_{xy}\epsilon$ in $\mathcal{O}(\log 1/\epsilon)$ *inner* iterations.

Convex case To achieve sublinear convergence of the *outer* iterations for convex \bar{f} , we need $\|e_k\| = \mathcal{O}(1/k^{1+\delta})$ for any $\delta > 0$ [15, Prop. 1]. To achieve this, we need to decrease the inner tolerance at the same rate: $\epsilon_k = \mathcal{O}(1/k^{1+\delta})$. Due to linear convergence of the inner iterations, this requires $\mathcal{O}(\log k)$ inner iterations. A total of K outer iterations thus has an asymptotic complexity of $\mathcal{O}(K \log K)$. Due to sublinear convergence of the outer iterations, we require $K = \mathcal{O}(1/\bar{\epsilon})$ outer iterations, giving a complexity of $\mathcal{O}(1/\bar{\epsilon} \log 1/\bar{\epsilon})$.

Strongly convex case To achieve linear convergence of the *outer* iterations for strongly convex \bar{f} , we need $\|e_k\| = \mathcal{O}(\gamma^k)$ for any $\gamma < 1$ [15, Prop. 3]. To achieve this, we need to decrease the inner tolerance at the same rate: $\epsilon_k = \mathcal{O}(\gamma^k)$. Due to linear convergence of the inner iterations, this requires $\mathcal{O}(k)$ inner iterations. A total of K outer iterations then require $\mathcal{O}(K^2)$ inner iterations. Due to linear convergence of the outer iterations, we require $K = \mathcal{O}(\log 1/\bar{\epsilon})$ outer iterations, giving an overall complexity of $\mathcal{O}((\log 1/\bar{\epsilon})^2)$.

2.4. Practical implementation. A basic proximal gradient method for solving problems of the form (1.1) is shown in Algorithm 2.1, where we denote

$$r \left(\begin{bmatrix} x \\ y \end{bmatrix} \right) := r_1(x) + r_2(y).$$

Algorithm 2.1 Prox-gradient for (1.1).

Require: Initial iterates, x_0, y_0 , Lipschitz constant, L , of ∇f .

$\alpha = 1/L$

$k = 0$

while $\|x_k - x_{k-1}\| + \|y_k - y_{k-1}\| > \bar{\epsilon}$ **do**
 $\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \text{prox}_{\alpha r} \left(\begin{bmatrix} x_k \\ y_k \end{bmatrix} - \alpha \nabla f \left(\begin{bmatrix} x_k \\ y_k \end{bmatrix} \right) \right)$
 $k = k + 1$

end while

The conventional VP algorithm with accurate inner solves is given in Algorithm 2.2. By Lemma E.1, the stopping criterion for the inner iterations guarantees that at outer iteration k , we have $\|y_l - \bar{y}(x_k)\| \leq (2L_{yy}/\mu)\epsilon$ and hence that the error in the gradient is bounded by

$$\|\nabla f(x_k, \bar{y}(x_k)) - \nabla f(x_k, y_l)\| \leq (2L_{xy}L_{yy}/\mu) \epsilon.$$

We use warmstarts for the inner iterations, which is expected to dramatically reduce the required number of inner iterations, especially when getting close to the solution.

An inexact version of Algorithm 2.2 can be implemented by specifying a decreasing sequence of tolerances $\{\epsilon_k\}_k$. However, in practice, it would be hard to figure out exactly how fast to decrease the tolerance. Inspired by the requirement for convergence of the inexact proximal gradient method, we therefore propose a stopping criterion based on the progress in the outer iterations. Theorem 2.5 requires that the error in the gradient $\|e_k\|$ is bounded by $(\bar{L}/2)\|x_{k+1} - x_k\|$. We can guarantee this by combining various bounds. By (2.2) and Lemma E.1, we get

$$\|y_l - y_{l-1}\| \leq \frac{\bar{L}\mu}{2L_{yy}L_{xy}} \|x_{k+1} - x_k\|,$$

Algorithm 2.2 Variable projection for (1.1) with accurate inner solves.

Require: Initial iterates, x_0, y_0 , Lipschitz constants \bar{L}, L_{yy} , tolerances $\epsilon, \bar{\epsilon}$.

$$\alpha = 1/\bar{L}$$

$$\beta = 1/L_{yy}$$

$$k = 0$$

while $\|x_k - x_{k-1}\| > \bar{\epsilon}$ **do**

$$l = 0$$

while $\|y_l - y_{l-1}\| > \epsilon$ **do**

$$y_{l+1} = \text{prox}_{\beta r_2}(y_l - \beta \nabla_y f(x_k, y_l))$$

$$l = l + 1$$

end while

$$y_0 = y_l$$

▷ Warmstart

$$x_{k+1} = \text{prox}_{\alpha r_1}(x_k - \alpha \nabla_x f(x_k, y_l))$$

$$k = k + 1$$

end while

which guarantees the required bound on the error. Note that x_{k+1} implicitly depends on y_l and x_k through $x_{k+1} = \text{prox}_{\alpha r_1}(x_k - \alpha \nabla_x f(x_k, y_l))$.

A practical implementation of the inexact method is shown in Algorithm 2.3. In practice, the constants \bar{L}, L_{yy}, L_{xy} , and μ may difficult to estimate. Moreover, the bounds may be very loose. We therefore introduce a parameter, ρ , to use in the stopping criterion instead.

In particular settings, the structure of the inner problem in y can be exploited to achieve superlinear convergence of the inner iterations. A further improvement could be to use a line search or an accelerated proximal gradient method for the outer iterations. However, accelerated proximal gradient methods are generally more sensitive to errors or require more information on the function, as pointed out by [15].

Algorithm 2.3 Algorithm for (1.1) with adaptive tolerance for the inner solves.

Require: Initial iterates, x_0, y_0 , Lipschitz constants \bar{L}, L_{yy} , parameter ρ

$$\alpha = 1/\bar{L}$$

$$\beta = 1/L_{yy}$$

$$k = 0$$

while $\|x_k - x_{k-1}\| > \bar{\epsilon}$ **do**

$$l = 0$$

repeat

$$x_{k+1} = \text{prox}_{\alpha r_1}(x_k - \alpha \nabla_x f(x_k, y_l))$$

▷ Prospective update of x_k

$$y_{l+1} = \text{prox}_{\beta r_2}(y_l - \beta \nabla_y f(x_k, y_l))$$

▷ Update of y_l

$$l = l + 1$$

until $\|y_l - y_{l-1}\| \leq \rho \|x_{k+1} - x_k\|$

$$y_0 = y_l$$

▷ Warmstart

$$k = k + 1$$

end while

3. Case studies.

3.1. Reproducibility. The Python code used to conduct the numerical experiments is available at <https://github.com/TristanvanLeeuwen/VarProNS>. We imple-

TABLE 3.1
Some examples of exponential data fitting in applications.

Application	ϕ_{ij}
Pharmacokinetic modelling	$x_i t_j$
Multiple signal classification	$x_i h_j$
Radial basis function interpolation	$\alpha_i^2 \ x_i - \xi_j\ ^2$

mented Algorithm 2.1 (hereafter referred to as the *joint* approach), Algorithm 2.2 (*VP*), and Algorithm 2.3 (*adaptive VP*). Instead of the absolute tolerance for the inner iterations in Algorithm 2.2, we use a relative tolerance and stop the inner iterations in Algorithm 2.2 when $\|y_l - y_{l-1}\| \leq \epsilon \|y_{l-1}\|$. For the outer iterations, we use a stopping criterion on the function value. The Lipschitz constants L, \bar{L}, L_{yy} and other algorithmic parameters are specified for each example. We show results for several values of ρ to investigate the sensitivity of the results to this parameter.

Aside from the example-specific results, we report the convergence history (in terms of the value of the objective) as a function of the number of outer iterations and the total number of iterations. The latter is used as an indication of the computational cost and will be referred to as *cost*.

3.2. Exponential data fitting. We begin with the general class of *exponential data-fitting* problems—one of the prime applications of variable projection [11]. The general formulation of these problems assumes a model of the form

$$d_i = \sum_{j=1}^n y_j \exp(-\phi_{ij}(x)), \quad i = 1, 2, \dots, m,$$

where $y \in \mathbb{R}^n$ are unknown weights, $d \in \mathbb{R}^m$ are the measurements, and ϕ_{ij} are given functions that depend on an unknown parameter $x \in \mathbb{R}^n$. Some examples of this class are given in Table 3.1.

The exponential data-fitting problem is traditionally formulated as a least-squares problem

$$\min_{x,y} \|A(x)y - d\|^2.$$

In many applications, however, it is natural to include regularization terms and/or use another data fidelity term. For example, [6, 16] consider positivity constraints $y_i \geq 0$. Another common regularization that enforces sparsity of y is $r_2(\cdot) = \lambda \|\cdot\|_1$. This is useful in cases where the system is overparametrized and we are looking for a fit of the data with as few components as possible. Even with regularization, we expect a highly ill-posed problem where many parameter combinations lead to a nearly identical data fit.

For the conditions of the previous theorems to hold, we require $A(x)^T A(x)$ to be invertible for all x . Note that this would require $x_i \neq x_j$ for $i \neq j$. As long as we initialize the x_i to be different, we do not expect any problems. Alternatively, a small quadratic term in y could be added to ensure that $f(x, y) = \|A(x)y - d\|^2 + \delta \|y\|^2$ is strongly convex in y .

3.2.1. Example 1. We model the data via a normal distribution with mean $\bar{d}_i = \sum_{j=1}^n y_j \exp(-t_i x_j)$ and variance σ . To generate the data, we set $m = 11, n = 2, t_i = (i - 1)/2, x = (0.1, 1.5), y = (2.1, 1.9)$, and $\sigma = 0.1$. Since the number of components, n , is not typically known in practice, we attempt to fit the data using

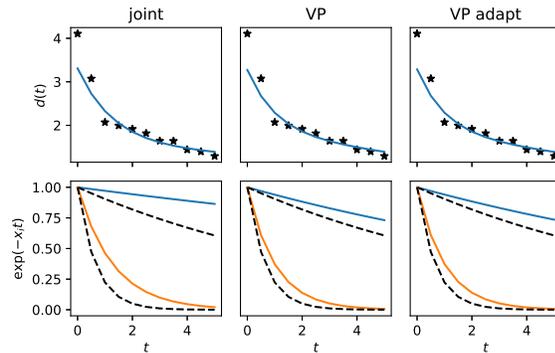


FIG. 3.1. *Top: data and fitted data resulting from the various methods. Bottom: individual components $\exp(-x_i t)$ resulting from the optimization as well as the ground-truth components (black dashed line).*

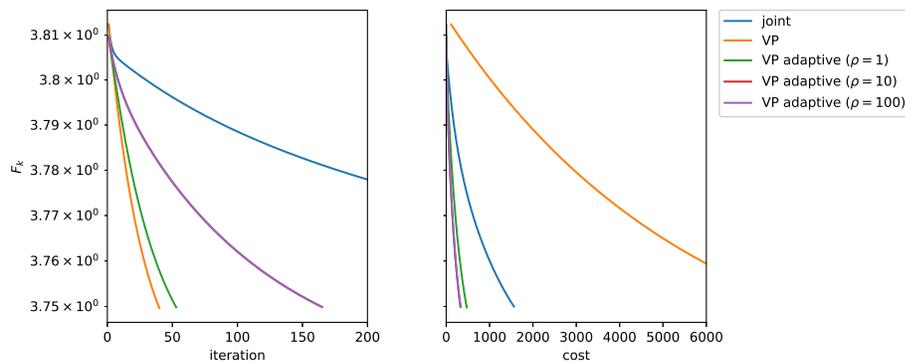


FIG. 3.2. *Convergence history and computational cost for Algorithms 1.1, 2.1, and 2.3. Note that the inexact VP algorithm retains the same favorable rate of convergence of the full VP algorithm while being significantly cheaper than the two alternatives.*

$n = 5$ components and use an ℓ_1 -regularization term with $\lambda = 1$ (chosen by trial and error) to find a parsimonious solution. The variational problem is then given by

$$\min_{x,y} \|A(x)y - d\|^2 + \lambda \|y\|_1 \quad \text{s.t.} \quad y_i \geq 0 \text{ for } i = 1, 2, \dots, n.$$

Here, $A(x)$ is an $m \times n$ matrix with elements $\exp(-t_i x_j)$. The Lipschitz constants of f and \bar{f} are set to $L = \bar{L} = 1 \cdot 10^3$ (selected by trial and error). For the inner solves, we let $L_{yy} = \|A(x)\|^2$ (for the true x), $\epsilon = 10^{-6}$ for VP, and $\rho \in \{1, 10, 100\}$ for adaptive VP. With these settings, we run Algorithm 2.1 (joint), Algorithm 2.2 (VP), and Algorithm 2.3 (adaptive VP). The results are shown in Figures 3.1 and 3.2. The VP approach gives a slightly better recovery of the modes $\exp(-x_i t)$ while achieving the same level of data fit as the joint approach. This indicates that the ill-posedness of the problem is effectively dealt with by the VP approach. The VP approach converges much faster than the joint approach. While the exact VP approach is much more expensive than the joint approach, the adaptive VP method is much cheaper than the joint approach.

3.2.2. Example 2. Here, we model the data via a Poisson distribution with parameter $\bar{d}_i = \sum_{j=1}^n y_j \exp(-t_i x_j)$. We set $m = 11$, $n = 2$, $t_i = (i - 1)/2$, $x =$

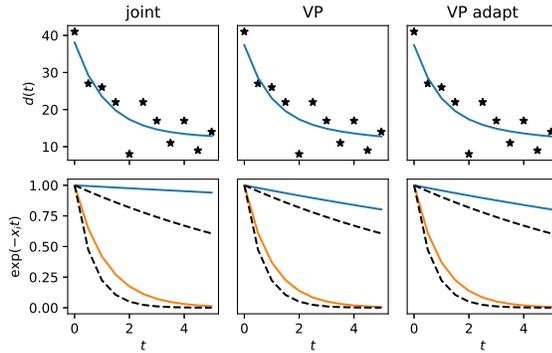


FIG. 3.3. *Top: data and fitted data resulting from the various methods. Bottom: individual components $\exp(-x_i t)$ resulting from the optimization as well as the ground-truth components (black dashed line).*

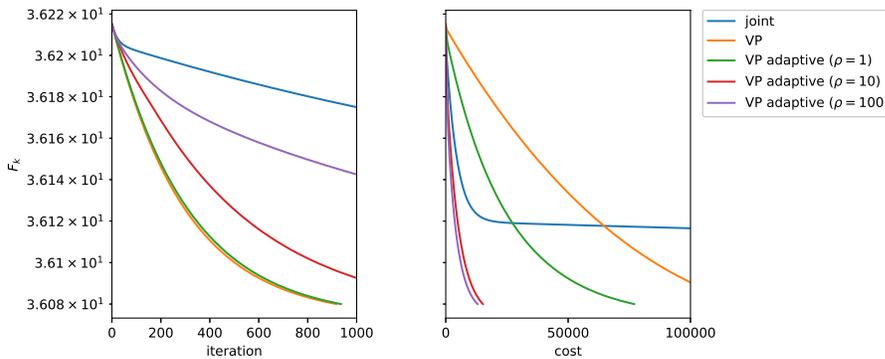


FIG. 3.4. *Convergence history and computational cost for Algorithms 2.1, 2.2, and 2.3. Note that the inexact VP algorithm retains the same favorable rate of convergence of the full VP algorithm while being significantly cheaper than the two alternatives.*

$(0.1, 1.5)$, $y = (21, 19)$. The other settings are exactly the same as the previous example. The variational problem is now given by

$$\min_{x, y} \ell(A(x)y, d) + \lambda \|y\|_1 \quad \text{s.t.} \quad y_i \geq 0 \text{ for } i = 1, \dots, n,$$

where ℓ denotes the Poisson log-likelihood function. The Lipschitz constants of f and \bar{f} are set to $L = \bar{L} = 5 \cdot 10^4$ (selected by trial and error). For the inner solves we let $L_{yy} = \|A(x)\|^2$ (for the true x), $\epsilon = 10^{-6}$ for inexact VP and $\rho \in \{1, 10, 100\}$ for adaptive VP. With these settings we run Algorithm 2.1 (joint), Algorithm 2.2 (VP), and Algorithm 2.3 (adaptive VP). The results are shown in Figures 3.3 and 3.4. Here again, the VP approach gives a slightly better recovery of the modes $\exp(-x_i t)$ while achieving the same level of data fit as the joint approach. This indicated the ill-posedness of the problem that is effectively dealt with by the VP approach. The VP approach converges much faster than the joint approach. The adaptive VP method is (much) cheaper than the joint approach, and even the regular VP approach eventually beats the joint approach.

3.3. Trimmed robust formulations in machine learning. Many formulations in high-dimensional regression, machine learning, and statistical inference can

be formulated as minimization problems

$$\min_x \sum_{i=1}^n \ell_i(x),$$

where the training set comprises n examples and ℓ_i is the error or negative log-likelihood corresponding to the i th training point. This approach can be made robust to perturbations of input data (for example, incorrect features, gross outliers, or flipped labels) using a *trimming approach*. The idea, first proposed by [13] in the context of least-squares fitting, is to minimize the $k \leq n$ best residuals. The general trimmed approach, formulated and studied by [20], considers the equivalent formulation

$$\min_{x,y} \sum_{i=1}^n y_i \ell_i(x), \quad y \in \hat{\Delta}_k,$$

where $\hat{\Delta}_k := \{y \in [0,1]^n : 1^T y = k\}$ denotes the *capped simplex* and admits an efficient projection [19, 1]. Jointly solving for (x, y) selects the k inliers as the model x is fit. Indeed, the reader can check that the solution in y for fixed x selects the smallest k terms f_i . The problem therefore looks like a good candidate for VP, but the projected function $\bar{f}(x)$ is nonsmooth because the solution for y may not be unique.¹ However, the smoothed formulation

$$(3.1) \quad \min_{x,y} \sum_{i=1}^n y_i \ell_i(x) + \frac{\delta}{2} \|y\|^2, \quad y \in \hat{\Delta}_k,$$

does lead to a differentiable $\bar{f}_\delta(x)$, as the problem is now strongly convex in y .

To illustrate the trimming method and the potential benefit of the VP approach, we consider the following stylized example.

3.3.1. Example 1. We aim to compute the (trimmed) mean of a set of samples $\{d_i\}_{i=1}^n$ by setting $\ell_i(x) = \frac{1}{2} \|x - d_i\|^2$. We generate data by sampling, generating a total of $n = 1000$ samples $d_i \in \mathbb{R}^2$; $k = 200$ are drawn from a normal distribution with mean $(-1, -1)$ and variance 1, while the remaining 800 are drawn from a normal distribution with mean $(1, 1)$ and variance 0.5. We let $\delta = 1 \cdot 10^{-3}$, $L = \bar{L} = k$ and $L_{yy} = \delta$, $\epsilon = 10^{-6}$ for VP and $\rho = 1$ for adaptive VP. The results for Algorithm 2.1 (joint), Algorithm 1.1 (VP), and Algorithm 2.3 (adaptive VP) are shown in Figures 3.5 and 3.6. We note that the VP approach gives a better recovery of the inliers. Moreover, the VP approach converges much faster than the joint approach. As the inner problem is a simple quadratic, both the exact and the adaptive VP approach have essentially the same computational cost.

3.4. Tomography. In computed tomography, the data consist of a set of line integrals of an unknown function

$$d_{ij} = \int_{\ell_{ij}} u(t) dt,$$

where ℓ_{ij} represents a line with angle θ_i and offset o_j . Collecting measurements for angles $\{\theta_i\}_{i=1}^k$ and offsets $\{o_j\}_{j=1}^m$ and representing the function u as a piecewise continuous function on a grid of $n \times n$ rectangular pixels leads to a system of linear

¹Consider, for example, a case where a number of residuals have the same value.

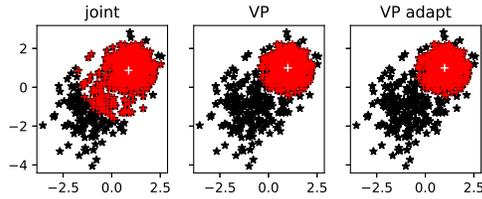


FIG. 3.5. Samples d_i (black), selected inliers (red), and estimated mean (white) for each method.

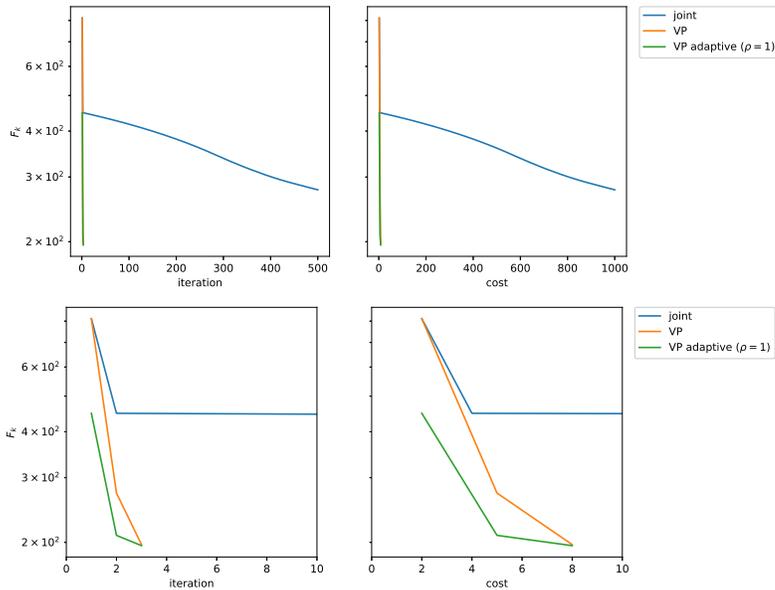


FIG. 3.6. Convergence history and computational cost for Algorithms 2.1, 1.1, and 2.3. The adaptive VP algorithm retains the same favorable rate of convergence of the full VP algorithm while being significantly cheaper than the two alternatives. The bottom plots show more clearly what happens in the first few iterations.

equations $d = Ay$, with $A \in \mathbb{R}^{m \cdot k \times n^2}$, with y_i representing the gray value of the image in the i th pixel. In many applications, the angles and absolute offsets are not known exactly due to calibration issues. To model this, we include additional parameters $x_i = (\Delta\theta_i, \Delta s_i)$ for each angle. This leads to the signal model

$$d = A(x)y + e,$$

where $x \in \mathbb{R}^{2k}$ contains the calibration parameters and e represents measurement noise.² We can ensure that $A(x)$ depends smoothly on x by using higher-order interpolation [18]. Many alternative methods have been proposed to solve the calibration problem [18, 3], and we do not claim that the proposed VP approach is superior; we merely use this problem as an example to compare the joint, VP, and adaptive VP approaches with each other.

²In low-dose applications, a Poisson noise model is more realistic.

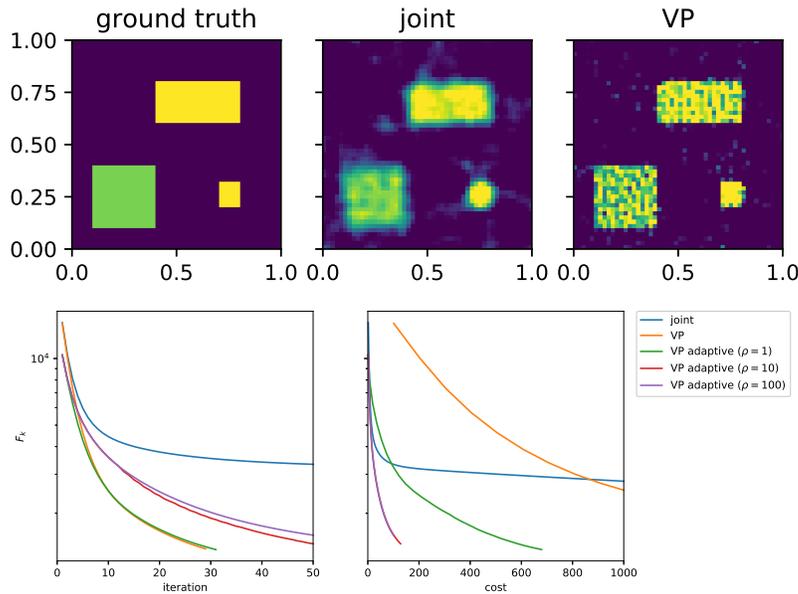


FIG. 3.7. *Top: ground-truth and reconstructed images. Bottom: convergence history and computational cost for Algorithms 2.1, 2.2, and 2.3. The inexact VP algorithm retains the same favorable rate of convergence of the full VP algorithm while being significantly cheaper than the two alternatives.*

3.4.1. Example 1. We let $n = 50$ and take $m = 50$ offsets regularly sampled in $[-1.5, 1.5]$ and $k = 50$ angles regularly sampled in $[0, 2\pi]$. This does not guarantee an invertible Hessian $A(x)^T A(x)$ (e.g., strongly convex $f(x, \cdot)$) for all x ; however, we will see that this does not lead to numerical issues in this particular example. As a safeguard, one could add a small quadratic regularizer $\delta \|y\|^2$ to the objective.

The data are generated with an additional random perturbation on the offset and angles (both are normally distributed with zero mean and variance 0.5) and measurement noise (normally distributed with zero mean and variance 1).

To regularize the problem, we include positivity constraints:

$$\min_{x,y} \|A(x)y - d\|^2 \quad \text{s.t.} \quad y_i \geq 0.$$

For the optimization, we let $L = \bar{L} = 1 \cdot 10^6$ and $L_{yy} = \|A(x)\|^2$ (for the true x), $\epsilon = 10^{-6}$ for VP, and $\rho \in \{1, 10, 100\}$ for adaptive VP. With these settings, we run Algorithm 2.1 (joint), Algorithm 2.2 (VP), and Algorithm 2.3 (adaptive VP). The results are shown in Figure 3.7. The VP approach gives a much better reconstruction than the joint approach due to the much faster convergence. The adaptive VP methods are cheaper than the joint method, again because they converge much faster. The adaptive VP approach, in turn, is much cheaper than the exact VP approach.

3.4.2. Example 2. The settings are the same as in the previous example, except that we use a TV-regularization term,

$$\min_{x,y} \|A(x)y - d\|^2 + \lambda \text{TV}(y),$$

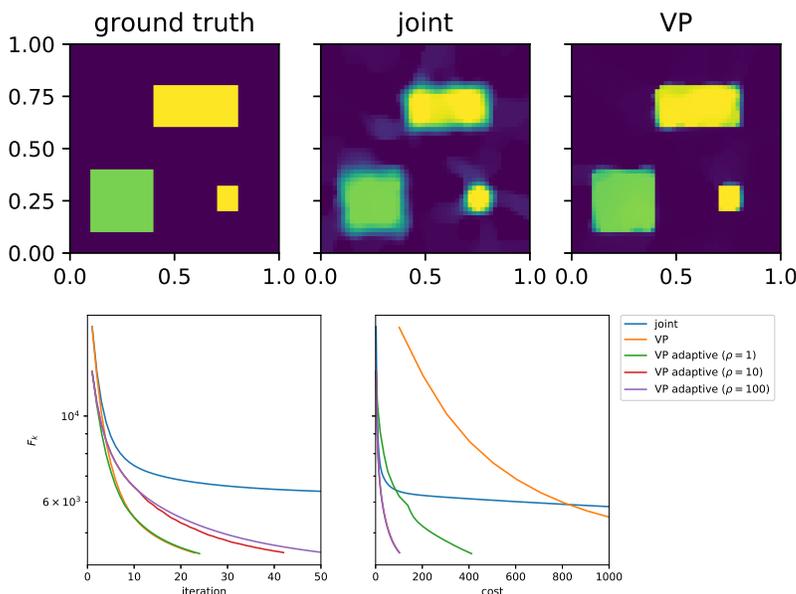


FIG. 3.8. *Top: ground-truth and reconstructed images. Bottom: convergence history and computational cost for Algorithms 2.1, 2.2, and 2.3. Note that the inexact VP algorithm retains the same favorable rate of convergence of the full VP algorithm while being significantly cheaper than the two alternatives.*

with

$$\text{TV}(y) = \sum_i \sqrt{(D_1 y)_i^2 + (D_2 y)_i^2},$$

where $D_k y$ returns a finite-difference approximation of the first derivative of the image y in the k^{th} direction. We let $\lambda = 2 \cdot 10^1$ (chosen by trial and error). The results are shown in Figure 3.8. The VP approach gives a much better reconstruction than the joint approach due to the much faster convergence. Both exact and adaptive VP are cheaper than the joint method, again because they converge much faster. The adaptive VP approach, in turn, is much cheaper than the exact VP approach.

4. Conclusions. Variable projection has been successfully used in a variety of contexts; the popularity of the approach is largely due to its superior numerical performance when compared to joint optimization schemes. In this paper, we extend its use to wide class of nonsmooth and constrained problems occurring in various applications. In particular, we give sufficient conditions for the applicability of an inner-outer proximal gradient method. We also propose an inexact algorithm with an adaptive stopping criterion for the inner iterations and show that it retains the same convergence rate as the exact algorithm. Numerical examples on a wide range of nonsmooth applications show that (i) the variable projection approach leads to faster convergence than the joint approach and that (ii) the adaptive variable projection method outperforms both the joint method and the exact variable projection method in terms of computational cost. The adaptive method includes a parameter, ρ , that controls the stopping tolerance for the inner iterations. In the numerical experiments, we observe that a larger value for ρ leads to a smaller number of outer iterations. There appears to be little danger of setting ρ too large; even when varying it over two

orders of magnitude, the adaptive VP method is consistently faster than the exact VP method. As $\rho \rightarrow 0$, the adaptive method coincides with the exact method. Heuristics could be developed to set this parameter automatically, but since this will be highly application specific, it is outside the scope of this paper.

Appendix A. Proof of Lemma 2.1.

Proof. Denote $F(x, y) = f(x, y) + r_2(y)$. For ease of notation, we fix x, x' and denote the corresponding (unique) optimal solutions by \bar{y} and \bar{y}' , respectively. These optimal solutions are implicitly defined by the first-order optimality conditions $0 \in \partial_y F(x, \bar{y})$ and $0 \in \partial_y F(x', \bar{y}')$ or, equivalently,

$$(A.1) \quad -\nabla_y f(x, \bar{y}) \in \partial r_2(\bar{y}), \quad -\nabla_y f(x', \bar{y}') \in \partial r_2(\bar{y}').$$

We start from strong convexity of F . For any \tilde{x} , we have

$$(A.2) \quad \mu \|\bar{y}' - \bar{y}\|^2 \leq \langle g'_{\tilde{x}} - g_{\tilde{x}}, \bar{y}' - \bar{y} \rangle,$$

with $g_{\tilde{x}} \in \partial_y F(\tilde{x}, \bar{y})$ and $g'_{\tilde{x}} \in \partial_y F(\tilde{x}, \bar{y}')$. We note that $\partial_y F(\tilde{x}, \bar{y}) = \{\nabla_y f(\tilde{x}, \bar{y}) + h \mid h \in \partial r_2(\bar{y})\}$, so we can write

$$\begin{aligned} g_{\tilde{x}} &= \nabla_y f(\tilde{x}, \bar{y}) + h, & h &\in \partial r_2(\bar{y}), \\ g'_{\tilde{x}} &= \nabla_y f(\tilde{x}, \bar{y}') + h', & h' &\in \partial r_2(\bar{y}'), \end{aligned}$$

with (A.2) holding for any choice of h and h' . Using (A.1), we make the particular choices $h = -\nabla_y f(x, \bar{y}) \in \partial r_2(\bar{y})$ and $h' = -\nabla_y f(x', \bar{y}') \in \partial r_2(\bar{y}')$. From (A.2), we now have

$$\mu \|\bar{y}' - \bar{y}\|^2 \leq \langle \nabla_y f(\tilde{x}, \bar{y}') - \nabla_y f(x', \bar{y}'), \bar{y}' - \bar{y} \rangle + \langle \nabla_y f(x, \bar{y}) - \nabla_y f(\tilde{x}, \bar{y}), \bar{y}' - \bar{y} \rangle.$$

Setting $\tilde{x} = x$, we get

$$\mu \|\bar{y}' - \bar{y}\|^2 \leq \langle \nabla_y f(x, \bar{y}') - \nabla_y f(x', \bar{y}'), \bar{y}' - \bar{y} \rangle.$$

Finally, using Cauchy–Schwarz and the Lipschitz smoothness of f in y , we have

$$\|\bar{y}' - \bar{y}\| \leq (L_{xy}/\mu) \|x' - x\|. \quad \square$$

Appendix B. Proof of Theorem 2.2.

Proof. We set out to show that

$$(B.1) \quad \lim_{\|e\| \rightarrow 0} \frac{|\bar{f}(x+e) - \bar{f}(x) - \nabla_x f(x, \bar{y}(x)) \cdot e|}{\|e\|} = 0,$$

which would confirm that $\nabla_x f(x, \bar{y}(x))$ is indeed the gradient of $\bar{f}(x)$.

Using the definition of \bar{f} , we rewrite (B.1) as

$$\lim_{\|e\| \rightarrow 0} \frac{|F(x+e, \bar{y}(x+e)) - F(x, \bar{y}(x)) - \nabla_x f(x, \bar{y}(x)) \cdot e|}{\|e\|}.$$

Writing $F(x+e, \bar{y}(x+e)) = F(x, \bar{y}(x+e)) + \nabla_x f(\xi, \bar{y}(x+e))$ with $\xi = x + te$ for $t \in [0, 1]$, we get

$$(B.2) \quad \lim_{\|e\| \rightarrow 0} \frac{|F(x, \bar{y}(x+e)) + \nabla_x f(\xi, \bar{y}(x+e)) \cdot e - F(x, \bar{y}(x)) - \nabla_x f(x, \bar{y}(x)) \cdot e|}{\|e\|}.$$

We now set out to bound the terms in the numerator of (B.2) in terms of $\|e\|^2$:

- For the gradient terms, we get (by Cauchy–Schwarz)

$$|(\nabla_x f(\xi, \bar{y}(x+e)) - \nabla_x f(x, \bar{y}(x))) \cdot e| \leq \|\nabla_x f(\xi, \bar{y}(x+e)) - \nabla_x f(x, \bar{y}(x))\| \cdot \|e\|.$$

Furthermore, by Lipschitz continuity of $\nabla_x f$, we have

$$\|\nabla_x f(\xi, \bar{y}(x+e)) - \nabla_x f(x, \bar{y}(x))\| \leq L_{xx}\|\xi - x\| + L_{xy}\|\bar{y}(x+e) - \bar{y}(x)\|.$$

Using that $\xi = x + te$ and Lipschitz continuity of \bar{y} (Lemma 2.1), we find

$$\|\nabla_x f(\xi, \bar{y}(x+e)) - \nabla_x f(x, \bar{y}(x))\| \leq (L_{xx} + L_{xy}L_{yx}/\mu)\|e\|.$$

This results in

$$|(\nabla_x f(\xi, \bar{y}(x+e)) - \nabla_x f(x, \bar{y}(x))) \cdot e| \leq (L_{xx} + L_{xy}L_{yx}/\mu)\|e\|^2.$$

- Next, we need to bound $|F(x, \bar{y}(x+e)) - F(x, \bar{y}(x))|$ in terms of $\|e\|^2$. Since $\bar{y}(x)$ is the optimal solution at x , we have $|F(x, \bar{y}(x+e)) - F(x, \bar{y}(x))| = F(x, \bar{y}(x+e)) - F(x, \bar{y}(x))$. Using the convexity of $F(x, \cdot)$, we have

$$F(x, \bar{y}(x+e)) - F(x, \bar{y}(x)) \leq \langle u, \bar{y}(x+e) - \bar{y}(x) \rangle,$$

with $u \in \partial_y F(x, \bar{y}(x+e))$. This can be expressed as $u = \nabla_y f(x, \bar{y}(x+e)) + w$ with $w \in \partial r_2(\bar{y}(x+e))$. Choosing w appropriately as $w = -\nabla_y f(x+e, \bar{y}(x+e))$, we get

$$\begin{aligned} & F(x, \bar{y}(x+e)) - F(x, \bar{y}(x)) \\ & \leq \|\nabla_y f(x, \bar{y}(x+e)) - \nabla_y f(x+e, \bar{y}(x+e))\| \|\bar{y}(x+e) - \bar{y}(x)\|. \end{aligned}$$

Using Lipschitz continuity of $\nabla_y f$ and \bar{y} , we get

$$F(x, \bar{y}(x+e)) - F(x, \bar{y}(x)) \leq (L_{yx}L_{xy}/\mu)\|e\|^2.$$

We have thus upper-bounded the term in (B.2) in terms of $\|e\|$. As this upper bound tends to zero as $\|e\| \rightarrow 0$ and the fraction is always nonnegative, we conclude that the limit tends to zero. \square

Appendix C. Proof of Corollary 2.3.

Proof. For ease of notation we fix x, x' and denote the corresponding (unique) optimal solutions by \bar{y} and \bar{y}' respectively. Using the definition $\nabla f(x) = \nabla_x f(x, \bar{y})$ we have

$$\|\nabla_x f(x, \bar{y}) - \nabla_x f(x', \bar{y}')\| \leq \|\nabla_x f(x, \bar{y}) - \nabla_x f(x', \bar{y})\| + \|\nabla_x f(x', \bar{y}) - \nabla_x f(x', \bar{y}')\|.$$

Using Lipschitz continuity of $\nabla_x f$ and \bar{y} we immediately find

$$\|\nabla_x f(x, \bar{y}) - \nabla_x f(x', \bar{y}')\| \leq (L_{xx} + L_{xy}L_{yx}/\mu)\|x - x'\|. \quad \square$$

Appendix D. Proximal gradient with errors.

Consider solving

$$\min_x F(x),$$

with $F(x) = f(x) + g(x)$, where f is proper, closed, and L -Lipschitz smooth and g proper closed and convex. We consider an inexact proximal gradient method of the form

$$(D.1) \quad x_{k+1} = \text{prox}_{\alpha g}(x_k - \alpha(\nabla f(x_k) + e_k)).$$

For the following, we closely follow [4].

LEMMA D.1 (sufficient decrease of inexact proximal gradient). *The iteration (D.1) produces iterates that obey*

$$F(x_k) - F(x_{k+1}) \geq (\alpha^{-1} - L/2) \|x_{k+1} - x_k\|^2 - \langle e_k, x_{k+1} - x_k \rangle.$$

Proof. By the smoothness assumption, we have

$$f(x_{k+1}) \leq f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k \rangle + \frac{L}{2} \|x_{k+1} - x_k\|^2.$$

By [4, Thm. 6.39], we have

$$\langle x_k - \alpha \nabla f(x_k) - \alpha e_k - x_{k+1}, x_k - x_{k+1} \rangle \leq \alpha g(x_k) - \alpha g(x_{k+1}),$$

which yields

$$\langle \nabla f(x_k), x_{k+1} - x_k \rangle \leq -\alpha^{-1} \|x_{k+1} - x_k\|^2 - \langle e_k, x_{k+1} - x_k \rangle + g(x_k) - g(x_{k-1}).$$

Combining gives the result. \square

We let $\alpha < 2/L$ and state a simple corollary that ensures that we can get descent if $\|e_k\|$ is small enough.

COROLLARY D.2 (existence of small enough errors). *At any iterate, we can always take $\|e_k\|$ small enough to ensure descent, unless x_k is a stationary point.*

Proof. Introduce the prox-gradient mapping T :

$$T(x) = \underset{\alpha g}{\text{prox}}(x - \alpha \nabla f(x)).$$

If x_k is nonstationary, we know that $\gamma := \|T(x_k) - x_k\|$ is bounded away from 0. On the other hand, the function

$$h(e_k) := \left\| \underset{\alpha g}{\text{prox}}(x_k - \alpha \nabla f(x_k) - \alpha e_k) - x_k \right\|$$

is α -Lipschitz continuous since the norm and the prox map are both 1-Lipschitz continuous, and we have $h(0) = \|T(x_k) - x_k\|$ and $h(e_k) = \|x_{k+1} - x_k\|$. Therefore, if we take $\|e_k\| \leq \frac{\gamma}{2\alpha}$, we have

$$\|h(0) - h(e_k)\| \leq \alpha \|e_k\| \leq \frac{\gamma}{2}$$

and therefore

$$\|x_{k+1} - x_k\| - \gamma < \frac{\gamma}{2} \Rightarrow \|x_{k+1} - x_k\| > \frac{\gamma}{2}.$$

So, if we take $\|e_k\| \leq \min(\frac{\gamma}{2\alpha}, \frac{\gamma}{2(\alpha^{-1} - L/2)})$, we are guaranteed that $\|e_k\| < (\alpha^{-1} - L/2)\|x_{k+1} - x_k\|$, ensuring descent by Lemma D.1. \square

THEOREM D.3 (convergence of inexact proximal gradient, general case). *The iteration (D.1) with step size $\alpha = 1/L$ and errors obeying $\|e_k\| \leq C\|x_{k+1} - x_k\|$ with $C < L/2$ produces iterates for which*

$$\min_{k \in \{0, 1, \dots, n-1\}} \|x_{k+1} - x_k\| \leq A \sqrt{\frac{F(x_0) - F(x_*)}{n}},$$

with $A = \sqrt{\frac{2}{L-2C}}$.

From Lemma D.1, we have

$$\|x_{k+1} - x_k\|^2 \leq \frac{2}{L} ((F(x_k) - F(x_{k+1})) + \cos \theta_k \|e_k\| \|x_{k+1} - x_k\|).$$

Summing over k , we have

$$\sum_{k=0}^{n-1} \|x_{k+1} - x_k\|^2 \leq \frac{2}{L} (F(x_0) - F(x_*)) + \frac{2}{L} \sum_{k=0}^{n-1} \|e_k\| \|x_{k+1} - x_k\|.$$

Assuming that $\|e_k\| \leq C \|x_{k+1} - x_k\|$, we get

$$\sum_{k=0}^{n-1} \|x_{k+1} - x_k\|^2 \leq \frac{2}{L} (F(x_0) - F(x_*)) + \frac{2C}{L} \sum_{k=0}^{n-1} \|x_{k+1} - x_k\|^2,$$

so

$$\left(1 - \frac{2C}{L}\right) \sum_{k=0}^{n-1} \|x_{k+1} - x_k\|^2 \leq \frac{2}{L} (F(x_0) - F(x_*)).$$

Hence,

$$\sum_{k=0}^{n-1} \|x_{k+1} - x_k\|^2 \leq \left(1 - \frac{2C}{L}\right)^{-1} \frac{2}{L} (F(x_0) - F(x_*)).$$

Thus,

$$\min_{k \in \{0, 1, \dots, n-1\}} \|x_{k+1} - x_k\| \leq \sqrt{\frac{2(F(x_0) - F(x_*))}{(L - 2C)n}}.$$

Appendix E. A stopping criterion for prox-gradient descent. When solving

$$\min_x f(x) + g(x),$$

where f is L -Lipschitz smooth and $f + g$ is μ -strongly convex, with a proximal gradient method, we need a practical stopping criterion that can guarantee a certain distance to the optimal solution. The following bound is useful.

LEMMA E.1 (a stopping criterion for proximal gradient descent). *Introduce the prox-gradient mapping T and the gradient mapping, $G(x)$,*

$$T(x) = \operatorname{prox}_{\alpha g}(x - \alpha \nabla f(x)),$$

$$G(x) = \alpha^{-1}(x - T(x)),$$

where f is L -Lipschitz smooth and μ -strongly convex, g is convex, and $\alpha \in (0, 2/L)$. Define the fixed point of G by \bar{x} . Then

$$\|T(x) - \bar{x}\| \leq \frac{1 + \alpha L}{\mu} \|G(x)\|.$$

Proof. By strong convexity of $F = f + g$, we have

$$\mu \|T(x) - \bar{x}\|^2 \leq \langle d, T(x) - \bar{x} \rangle,$$

with $d \in \partial F(T(x))$. Note that $\partial F(x) = \nabla f(x) + \partial g(x)$ and that

$$\alpha^{-1}(x - T(x)) - \nabla f(x) \in \partial g(T(x))$$

by definition of the proximal operator. Picking this particular element in $\partial g(T(x))$, we obtain

$$\mu \|T(x) - \bar{x}\|^2 \leq \langle G(x), T(x) - \bar{x} \rangle + \langle \nabla f(T(x)) - \nabla f(x), T(x) - \bar{x} \rangle.$$

By Lipschitz-smoothness and Cauchy–Schwarz, we get

$$\|T(x) - \bar{x}\| \leq \frac{1 + \alpha L}{\mu} \|G(x)\|. \quad \square$$

Remark E.2. When applying a standard proximal gradient method $x^+ = T(x)$, we can immediately use this bound to devise a stopping criterion $\|x^+ - x\| \leq \epsilon$ that guarantees $\|x^+ - \bar{x}\| \leq \epsilon(\alpha^{-1} + L)/\mu$.

REFERENCES

- [1] A. ARAVKIN AND D. DAVIS, *A Smart Stochastic Algorithm for Nonconvex Optimization with Applications to Robust Machine Learning*, preprint, arXiv:1610.01101, 2016.
- [2] A. Y. ARAVKIN AND T. VAN LEEUWEN, *Estimating nuisance parameters in inverse problems*, *Inverse Problems*, 28 (2012), 115016.
- [3] A. P. AUSTIN, Z. WENDY, S. LEYFFER, AND S. M. WILD, *Simultaneous sensing error recovery and tomographic inversion using an optimization-based approach*, *SIAM J. Sci. Comput.*, 41 (2019), pp. B497–B521.
- [4] A. BECK, *First-Order Methods in Optimization*, SIAM, Philadelphia, 2017.
- [5] B. M. BELL AND J. V. BURKE, *Algorithmic differentiation of implicit functions and optimal values*, *Advances in Automatic Differentiation*, C. H. Bischof, H. M. Bücker, P. D. Hovland, U. Naumann, and J. Utke, eds., Springer, Berlin, 2008, pp. 67–77.
- [6] A. CORNELIO, E. L. PICCOLOMINI, AND J. G. NAGY, *Constrained variable projection method for blind deconvolution*, *J. Phys. Conf. Ser.*, 386 (2012), pp. 6–11.
- [7] O. DEVOLDER, F. GLINEUR, AND Y. NESTEROV, *First-order methods of smooth convex optimization with inexact oracle*, *Math. Program.*, 146 (2014), pp. 37–75.
- [8] G. H. GOLUB AND V. PEREYRA, *The differentiation of pseudo-inverses and nonlinear least squares whose variables separate*, *SIAM J. Numer. Anal.*, 10 (1973), pp. 413–432.
- [9] G. H. GOLUB AND V. PEREYRA, *Separable nonlinear least squares: The variable projection method and its applications*, *Inverse Problems*, 19 (2003), R1.
- [10] M. R. OSBORNE, *Separable least squares, variable projection, and the Gauss-Newton algorithm*, *Electron. Trans. Numer. Anal.*, 28 (2007), pp. 1–15.
- [11] V. PEREYRA AND G. SCHERER, EDs., *Exponential Data Fitting and Its Applications*, Bentham Science Publishers, Sharjah, United Arab Emirates, 2012.
- [12] R. T. ROCKAFELLAR AND R. J. B. WETS, *Variational Analysis*, Vol. 317, Springer, Berlin, 1998.
- [13] P. J. ROUSSEEUW, *Least median of squares regression*, *J. Amer. Statist. Assoc.*, 79 (1984), pp. 871–880.
- [14] A. RUHE AND P. A. WEDIN, *Algorithms for separable nonlinear least squares problems*, *SIAM Rev.*, 22 (1980), pp. 318–337.
- [15] M. SCHMIDT, N. L. ROUX, AND F. R. BACH, *Convergence rates of inexact proximal-gradient methods for convex optimization*, in *Advances in Neural Information Processing Systems*, 2011, pp. 1458–1466.
- [16] P. SHEARER AND A. C. GILBERT, *A generalization of variable elimination for separable inverse problems beyond least squares*, *Inverse Problems*, 29 (2013), 045003.
- [17] R. L. SMITH, *Some interlacing properties of the Schur complement of a Hermitian matrix*, *Linear Algebra Appl.*, 177 (1992), pp. 137–144.
- [18] T. VAN LEEUWEN, S. MARETZKE, AND K. J. BATENBURG, *Automatic alignment for three-dimensional tomographic reconstruction*, *Inverse Problems*, 34 (2018), 024004.
- [19] W. WANG AND C. LU, *Projection onto the Capped Simplex*, preprint, arXiv:1503.01002, 2015.
- [20] E. YANG, A. C. LOZANO, A. ARAVKIN, ET AL., *A general family of trimmed estimators for robust high-dimensional data analysis*, *Electron. J. Stat.*, 12 (2018), pp. 3519–3553.